After Human Mathematics

AI, Machine-Born Abstraction, and the Discovery of New Mathematical Worlds

Felix Mönnich

Douglas C. Youvan

Contents

Preface: Mathematics After the Mathematician

Part I: The Mathematics We Inherited

Chapter 1: The Historical Shape of Mathematical Thought

Chapter 2: Proof Is Not the Whole of Mathematics

Chapter 3: The Discovery of the Right Representation

Part II: When Mathematics Becomes Searchable

Chapter 4: Searching the Space of Definitions

Chapter 5: Counterexamples as a Creative Engine

Chapter 6: Experimental Mathematics at Machine Scale

Chapter 7: The Automatic Discovery of Abstraction

Part III: Machine-Born Mathematics

Chapter 8: Finding Hidden Equivalences

Chapter 9: Designing Entire Theories

Chapter 10: Mathematics as Compression

Chapter 11: Beyond Human Notation

Part IV: The New Mathematical World

Chapter 12: Self-Growing Mathematics

Chapter 13: Mathematics Without Historical Boundaries

Chapter 14: The Translation Problem

Chapter 15: After Human Mathematics

Epilogue: What Would Count as Discovery?

References

Preface

Mathematics After the Mathematician

For most of recorded history, mathematics has been inseparable from the limitations of the human mathematician. This is not a criticism of mathematics. It is simply a historical fact. A mathematical object had to be conceived by a human mind. A definition had to be short enough to state. A proof had to be divided into steps that a human reader could follow. A useful notation had to fit on paper, a blackboard, or a printed page. An abstraction had to be memorable enough to transmit from teacher to student. Even when calculations were enormous, the conceptual organization surrounding those calculations remained constrained by human attention, memory, intuition, language, education, and lifespan.

The mathematics we possess is therefore not necessarily the mathematics that exists. It is the mathematics that human beings have managed to find. The distinction matters. Artificial intelligence is usually discussed in mathematics as though it were another tool. Computers calculate. Symbolic systems manipulate expressions. Automated theorem provers verify deductions. Machine-learning systems suggest lemmas or search proof spaces. Large language models explain known mathematics and increasingly assist mathematicians with formalization, programming, conjecture generation, and proof. These are significant developments, but they may represent only the earliest phase. The more consequential possibility is that sufficiently capable artificial intelligence will cease merely operating inside mathematics created by humans.

It may begin changing the structure of mathematics itself. An advanced mathematical intelligence could search spaces that mathematicians rarely regard as searchable. It could search not merely for a proof of a theorem, but for the best representation of the problem. It could search among definitions. It could vary axioms. It could construct millions of finite mathematical worlds and identify recurring structures among them. It could propose abstractions, attempt to destroy them with counterexamples, repair them, and determine whether the repaired structures connect previously unrelated fields. Eventually it could search over entire mathematical theories. The transition would be subtle at first. A machine finds a shorter proof.

Then it discovers a lemma nobody thought to formulate. Then it proposes a useful invariant. Then a new definition. Then an unexpected equivalence

between theories. At some point the distinction between assisting mathematics and creating mathematics becomes difficult to maintain. The crucial event may not be the first machine proof of a difficult theorem. Proof, important as it is, represents only one stage in mathematical discovery. A much more profound event would occur when an artificial system says, in effect:

There is a concept missing from your mathematics.

It introduces the concept. Thousands of observations that had previously appeared unrelated become special cases. Several difficult proofs collapse into short arguments. Two fields thought to be separate become translations of one another. The machine has not merely answered a mathematical question. It has changed the questions mathematics is capable of asking. That would place artificial intelligence in a role traditionally occupied by the greatest mathematical innovators. Euclid did not merely solve geometry problems. He reorganized mathematical knowledge axiomatically. Descartes did not merely solve geometric problems. He changed geometry by introducing coordinates. Newton and Leibniz developed a language capable of treating continuous change. Galois transformed the study of equations by moving attention from explicit solutions to symmetry.

Riemann transformed geometry by loosening it from ordinary Euclidean space. Hilbert emphasized formal structure and axiomatization. Noether revealed structural invariants across algebra and physics. Grothendieck repeatedly replaced mathematical objects with richer relational worlds in which apparently difficult problems became natural. Category theory changed the emphasis yet again: from objects considered in isolation toward maps, transformations, and universal relationships. Each of these advances involved more than proving additional statements. They changed representation. They changed ontology. They changed what mathematicians considered to be a legitimate mathematical object. Artificial intelligence may eventually perform the same operation continuously. The result could be mathematics developing at a rate, scale, and conceptual depth impossible under human cognition alone.

This book explores that possibility. It is deliberately speculative. It does not assume that artificial general intelligence currently exists, nor does it depend on any particular contemporary AI architecture. The argument is more general. Suppose an artificial system eventually becomes capable of conducting mathematical research with considerable autonomy. What would it actually do? Would it merely solve our unsolved problems? Or would it discover that our problems are artifacts of how humans chose to represent mathematics? Would it retain our familiar divisions among algebra,

geometry, topology, logic, probability, and computation? Or would those divisions dissolve? Would it use our notation? Would it choose our axioms?

Would it regard proof in the same way we do? Would mathematical objects discovered by machines remain understandable to humans? And, perhaps most importantly:

What would mathematics look like if the cognitive limitations that shaped its historical development were no longer binding?

That question leads toward something stranger than automated theorem proving. It leads toward the possibility of machine-born mathematics.

PART I
The Mathematics We Inherited

CHAPTER 1

The Historical Shape of Mathematical Thought

Mathematics often presents itself as timeless. A theorem appears true independently of who proved it, when it was discovered, or which civilization first formulated it. Once we understand that there are infinitely many prime numbers, the proposition seems to stand outside history. Euclid discovered a proof, but he did not cause the theorem to become true. This timeless quality makes it easy to forget that mathematics as a human activity is intensely historical. The truths may not depend upon history. The route by which we encounter them does. Human mathematics has grown through a sequence of representational inventions. Numbers allowed collections to be abstracted from the things being counted. Algebra separated patterns of calculation from particular numerical instances. Coordinate systems translated geometry into equations. Calculus created machinery for reasoning systematically about change. Abstract algebra moved beyond calculation toward structure. Topology learned to disregard measurement in favor of continuity and deformation. Category theory increasingly described mathematical worlds through transformations rather than isolated objects.

None of these developments merely enlarged a database of known facts. Each created a new space of thinkable thoughts. Before an adequate notation exists, a mathematical relationship can be technically present while remaining conceptually inaccessible. Consider the difference between performing repeated arithmetic and writing an algebraic equation. The algebraic expression compresses infinitely many possible numerical instances into a single structure. Once the structure has been represented, operations that would otherwise require repeated reasoning become transformations of notation. Representation changes difficulty. This principle appears repeatedly throughout mathematics. A complicated geometric intersection becomes an algebraic system after coordinates are introduced. An apparently unrelated family of polynomial equations becomes organized by group structure.

An enormous collection of individual transformations becomes manageable once recognized as instances of a common morphism. A difficult global problem becomes tractable when decomposed into local pieces with compatibility conditions. A long calculation disappears when the correct invariant is discovered. Some mathematical breakthroughs are therefore better described as discoveries of representation than discoveries of facts. The

fact was already there. What changed was the language in which the fact became visible.

Mathematical Difficulty Is Representation-Dependent

This observation leads to a deceptively simple principle:

The difficulty of a mathematical problem is not solely a property of the problem. It is partly a property of its representation.

Humans know this intuitively. A novice may struggle to add Roman numerals that are trivial in decimal positional notation. A system of equations that appears cumbersome in one coordinate system may simplify dramatically after rotation or diagonalization. A recurrence relation can become transparent through a generating function. A geometric symmetry may be obscure in coordinates but immediate from a group action. A topological construction may appear pathological when described point by point yet become natural through a universal property. What we call ingenuity frequently consists of finding the representation in which the difficulty disappears. Yet traditional mathematics does not normally treat representation itself as a large-scale computational search problem.

A mathematician develops intuition. She tries a substitution. She draws a diagram. She recognizes that a problem resembles another one. She changes coordinates. She defines a new object. She notices symmetry. These activities are among the least formalized parts of mathematical creativity. Proof begins only after much of the deepest work has already occurred. This matters for artificial intelligence because machines need not inherit that division. An advanced system could treat the choice of representation as something to optimize systematically. Suppose a problem admits a thousand reasonable encodings. A human mathematician might examine five. A sufficiently capable machine might examine all thousand, generate another million, and develop metrics predicting which representations are likely to expose invariants, simplify proof search, or connect the problem to known structures.

This would change the role of mathematical intelligence. Instead of asking only:

How do I solve this problem?

the system could ask:

In what mathematical world does this problem become easy?

That is a much more powerful question.

The Contingency of Mathematical History

There is another consequence. The organization of mathematics into contemporary fields may not reflect a unique natural decomposition of mathematical reality. It also reflects history. Problems appeared in particular contexts. Techniques developed around them. Communities formed. Terminologies hardened. Journals, departments, textbooks, conferences, and educational sequences reinforced disciplinary boundaries. Algebra became algebra. Geometry became geometry. Probability became probability. Logic became logic. Topology became topology. These divisions are enormously useful, but there is no reason to assume they represent the final ontology of mathematics. From a sufficiently abstract point of view, boundaries repeatedly disappear. Geometry becomes algebraic. Algebra becomes geometric. Logic becomes categorical. Programs correspond to proofs.

Types acquire homotopical structure. Probability meets geometry. Dynamics meets topology. Combinatorics appears everywhere. The mathematical landscape seems less like a collection of separate continents than like a single terrain viewed through different coordinate systems. Humans discover those connections slowly. An artificial intelligence might search for them deliberately. Given enormous bodies of formalized mathematics, it could construct graphs whose nodes are definitions, theorems, constructions, examples, algorithms, and invariants. It could search for translations between regions. It could look for recurrent diagrams. It could identify proofs that are structurally identical despite different terminology. Eventually it might conclude that two subjects are not merely analogous. They are projections of the same deeper structure.

That possibility raises an uncomfortable thought. Some of the mathematics we regard as fundamental may be fundamental primarily because human beings encountered it early. Other structures, perhaps far more powerful, may remain obscure because they lie outside the natural pathways of human mathematical development. AI could explore those neglected regions.

Human Cognitive Bottlenecks

Human mathematics operates under constraints so familiar that we rarely notice them. Working memory is small. Attention is serial. Life is short. Education takes decades. Notation must remain legible. Definitions cannot contain millions of interacting clauses. Proofs must eventually be communicable. The number of examples a mathematician can deeply examine is limited. The number of alternative axiomatic systems she can explore is tiny. The number of candidate definitions she can seriously

evaluate is smaller still. These constraints shape mathematical taste. We favor elegance partly because elegance is cognitively efficient. We seek short definitions. We prize powerful abstractions. We value invariants because they allow enormous families of cases to be compressed into manageable ideas.

These preferences may reflect deep features of mathematics. But they also reflect the architecture of human cognition. A machine intelligence could share some of these preferences for entirely different reasons. Compression is computationally useful. Reusable abstractions reduce search. Symmetry reduces description length. Modular proofs simplify verification. But a machine would not necessarily have our thresholds. A mathematical object too complicated for any human to keep conceptually present might be perfectly natural to an artificial system. A theory requiring simultaneous awareness of ten thousand interacting transformations might be unusable by us and routine for it. Conversely, an artificial intelligence might eventually discover that extraordinarily complex human theories admit much simpler machine-native representations.

Either possibility destabilizes the assumption that human mathematical notation is close to mathematics itself.

From Calculation to Concept Formation

The history of computing in mathematics began largely with calculation. Machines could perform arithmetic faster than humans. Then symbolic systems manipulated algebraic expressions. Computers enabled massive numerical simulations. Proof assistants formalized deduction. Automated theorem provers searched logical spaces. Machine learning began predicting useful proof steps and mathematical patterns. Each stage moved closer to activities traditionally regarded as intellectual rather than mechanical. But concept formation remains a major frontier. Consider how a new mathematical concept emerges. A mathematician encounters several examples. Something appears recurrent. The examples differ superficially but behave similarly. A definition is proposed. The first definition fails. A counterexample appears. The definition is refined.

New examples fit. The concept interacts unexpectedly with existing theory. Eventually it becomes obvious in retrospect. Future students learn the definition in a paragraph and may never realize how difficult it was to invent. This process is potentially mechanizable in a much deeper sense than ordinary theorem proving. A machine could generate candidate abstractions from families of examples. It could score them according to compression, predictive usefulness, theorem-generating power, compositionality, robustness under perturbation, and connections with other structures. It could

search for counterexamples automatically. It could refine definitions. And it could retain successful abstractions as reusable components of an expanding mathematical ontology. The machine would no longer merely reason with concepts.

It would manufacture them.

The Missing Concept

Imagine an artificial system examining a large mathematical database. It notices that hundreds of theorems in different disciplines share an underlying pattern. Human authors have described the pattern differently because each theorem emerged from a different tradition. The system proposes a new object, call it X. It defines a class of X-structures and appropriate transformations between them. It then proves that several known constructions are special cases. A collection of previously difficult theorems follows from two general results about X. Entire sections of several subjects become instances of one theory. This would be mathematical discovery of a very high order. The discovery would not necessarily be a new theorem.

It would be a new **way for theorems to exist together**. Such moments have repeatedly transformed mathematics. An advanced artificial intelligence could potentially search for them continuously. The relevant objective might be something like:

Find the smallest new conceptual structure that makes the largest body of existing mathematics simpler.

That is a compression problem. It is also an abstraction problem. And it may become one of the central tasks of machine mathematics.

Mathematics as a Space Yet to Be Explored

We often speak metaphorically of mathematical discovery. A theorem is "found." A structure is "uncovered." A mathematician "explores" a space. The metaphors imply that mathematics possesses a landscape larger than our current knowledge. Artificial intelligence may make that metaphor operational. Instead of one mathematician exploring a narrow region over a career, machine systems could traverse huge populations of mathematical worlds. They could enumerate finite examples. Generate structures. Alter axioms. Construct morphisms. Calculate invariants. Search for universality. Detect equivalences. Reject barren theories. Expand fertile ones. What emerges would not simply be a larger collection of theorems. It could be an increasingly detailed map of the space of possible mathematics.

That map may look very different from our textbooks. The organization inherited from mathematical history might survive where it reflects genuine structural boundaries. Elsewhere it could dissolve. And beyond the familiar regions, there may be structures for which human mathematics currently has no names. The first great transformation brought by artificial intelligence to mathematics may therefore not be that machines prove theorems faster than mathematicians. It may be that machines begin exploring places mathematicians never thought to look. And once that happens, the historical shape of mathematics will cease to be the boundary of mathematics itself.

CHAPTER 2

Proof Is Not the Whole of Mathematics

Artificial intelligence has often been evaluated in mathematics by one question:

Can it prove the theorem?

The question is natural. Proof is the formal currency of mathematics. A mathematical claim becomes established not because it is plausible, elegant, frequently observed, numerically convincing, or endorsed by authority, but because it follows from accepted assumptions through valid reasoning. For this reason, automated theorem proving has occupied a central place in attempts to mechanize mathematics. If deduction can be formalized, perhaps a sufficiently capable machine can search the resulting logical space until it finds a valid proof. This program has already produced remarkable achievements. Computers verify enormous calculations. Proof assistants certify arguments with a degree of formal precision difficult to achieve in ordinary mathematical prose. Automated systems search for deductions beyond what humans would willingly perform by hand. Machine-learning methods increasingly guide proof search toward promising steps.

Yet proving a theorem after the theorem has already been formulated captures only a portion of mathematical intelligence. Often it captures the easier portion. The hardest step may have occurred earlier. Someone had to decide what the theorem should say. Someone had to identify the relevant objects. Someone had to choose the definitions. Someone had to notice which examples belonged together. Someone had to select a representation in which the desired property became visible. Someone had to recognize that a failed theorem could be repaired by modifying a hypothesis. Someone had to decide that an entirely new concept was needed. By the time a clean theorem arrives on the page, much of the intellectual struggle has disappeared.

Proof is the final visible artifact of a much larger process.

The Theorem Before the Proof

Consider a typical textbook theorem. It may be stated in three lines and proved in half a page. To the student, the central challenge is understanding the proof. But historically the theorem may have required years of conceptual development before anyone knew that this was the correct statement to prove.

A mathematician often begins with something far less tidy. There may be numerical evidence. Geometric intuition. A family of examples. A failed analogy. A conjecture that is almost true. Several counterexamples. A calculation that behaves unexpectedly. A connection to another area whose significance is not yet understood. The final theorem emerges through repeated refinement. Its hypotheses are adjusted.

Its language changes. Sometimes the original question disappears entirely. The result eventually becomes so natural that later generations forget how contingent its formulation once was. This is one reason mature mathematics can create a distorted picture of discovery. A polished paper presents mathematics in logical order. Definitions come first. Then lemmas. Then propositions. Then the main theorem. But the mathematics was rarely discovered in that order. The definition may have been invented after the theorem was suspected. The lemma may originally have been the main result. The final proof may have become obvious only after the right abstraction was found. The published sequence records logical dependence, not cognitive history.

An artificial intelligence trained primarily on finished mathematics therefore encounters a highly compressed residue of mathematical thought. It sees answers stripped of much of the search that produced them. If AI is to become genuinely creative in mathematics, it must eventually reconstruct that hidden process.

What Is the Problem?

Even before theorem formulation lies a more basic act:

deciding what deserves investigation.

Mathematical research is not an exhaustive attempt to prove every statement that can be written. The space of possible propositions is far too large. Most are uninteresting. Many are arbitrary. Some are technically difficult but conceptually barren. Others are easy yet unexpectedly illuminating. A mathematician develops a sense of promising questions. Why does this pattern persist? What is invariant? Can this construction be generalized? Is this hypothesis necessary? What happens in higher dimension? Why do two unrelated formulas have the same form? Can a local statement be made global? Does this phenomenon reflect symmetry? What breaks if commutativity is removed? Is there a universal object hiding here?

These questions guide attention before proof search begins. A mathematical AI capable only of proving supplied conjectures is therefore dependent on someone else to define its research agenda. A more

autonomous system would need to generate questions. That immediately introduces problems absent from ordinary theorem proving. How should the system distinguish an important question from a trivial one? What makes one conjecture worth pursuing and another disposable? How should novelty be balanced against tractability? How should elegance, explanatory power, generality, usefulness, and surprise be valued? These are not merely philosophical concerns. They become objective-function problems for machine mathematics. A mathematical intelligence needs some principle for allocating its finite computational resources.

It needs mathematical taste.

Mathematical Taste

Mathematicians use the word beautiful with unusual frequency for a technical discipline. A proof is beautiful. A theorem is elegant. A definition is natural. A construction is ugly. A hypothesis feels artificial. An argument is too computational. Another proof reveals the reason. These judgments are difficult to formalize, yet they play an important role in research. Two proofs may establish exactly the same proposition while contributing radically different understanding. One may consist of a massive case analysis. The other may reveal a symmetry that explains why the theorem had to be true. Both satisfy the logical requirement. Only one transforms our understanding. Likewise, two definitions may describe the same collection of examples while differing greatly in mathematical usefulness.

One definition may be an arbitrary list of conditions. Another may expose a universal property from which important consequences follow automatically. The second is often regarded as more natural. An advanced AI will need some analogue of this distinction if it is to move beyond mechanical deduction. It must prefer not merely correct mathematics but **productive mathematics**. That preference could potentially be learned. Concepts that compress many theorems are valuable. Definitions preserved under natural transformations may be valuable. Structures appearing independently across many domains may be valuable. Proofs that reduce description length or eliminate arbitrary choices may be valuable. Constructions that generate many new questions may be valuable.

An AI could develop quantitative approximations to mathematical taste by measuring such properties. Its taste might initially resemble ours. Eventually it might diverge.

Proof Versus Explanation

The distinction between proof and explanation is especially important. A proof answers:

Is this statement necessarily true under these assumptions?

An explanation attempts to answer:

Why should we have expected it to be true?

The two frequently overlap, but not always. A computer-generated proof could contain thousands of steps and be completely valid while providing little insight. Conversely, a short conceptual argument may reveal a structural reason that unifies many individual cases. Future mathematical AI could explicitly optimize for both. It might first find any proof. Then seek a more compressive proof. Then search for an invariant explaining the argument. Then generalize the invariant. Then recognize that the generalization belongs to a larger theory. The original theorem becomes merely an entry point. Consider a system that discovers that hundreds of computational identities are consequences of the same symmetry group.

It has done something more important than verifying the identities independently. It has replaced many facts with one reason. This operation - replacing a large body of mathematical facts by a smaller explanatory structure - may be one of the defining activities of advanced machine mathematics.

Counterexamples Are Part of Discovery

Proof is usually presented positively. A statement is proposed. Then it is established. But much mathematical creativity is driven by failure. Suppose a conjecture appears true for the first hundred examples. A counterexample is found. There are several possibilities. The conjecture may simply be wrong. Or the counterexample may reveal a missing hypothesis. Perhaps compactness is required. Perhaps finite dimension. Perhaps smoothness. Perhaps connectedness. Perhaps an orientation. Perhaps an assumption thought irrelevant turns out to be essential. A single counterexample can therefore teach more than a hundred confirming examples. It exposes the boundary of a concept. Advanced AI could turn this into an industrial-scale process.

Every conjecture it generates could immediately trigger a counterexample search. Finite structures could be enumerated. Random examples sampled. Symbolic constraints solved. Degenerate cases explored. Known pathological constructions imported. Alternative foundations tested. The system could

deliberately seek the smallest example that breaks the claim. This would create a cycle:

conjecture -> attack -> failure -> repair -> stronger conjecture.

The goal would not be to protect ideas from criticism. It would be to make criticism the mechanism by which ideas improve. In this sense, an automated mathematician might behave less like a theorem prover and more like a permanent adversarial collaborator.

Definitions Can Be Wrong Too

Mathematicians usually speak of theorems as true or false. Definitions occupy a different logical status. A definition cannot ordinarily be false in the same way. One can define any object one wishes. But definitions can certainly be bad. They can be useless. Overly broad. Overly narrow. Unstable. Redundant. Historically accidental. Dependent on arbitrary presentation. Poorly behaved under the transformations that matter. A sophisticated mathematical AI could treat definitions as hypotheses about how mathematical reality should be organized. Suppose many examples exhibit related behavior. The AI proposes a candidate definition D1. It studies the resulting class. The class is too large. Pathological examples dominate. It strengthens the definition to D2.

Now important examples disappear. It searches for another condition, producing D3. Under D3, the class becomes closed under useful operations, admits a clean notion of morphism, possesses meaningful invariants, and connects naturally with several existing theories. The definition has been discovered through optimization. This is a very different activity from proof search. The machine is exploring a landscape of possible concepts.

The Right Morphisms

There is an old lesson in abstract mathematics: understanding the objects is often not enough. One must also identify the correct maps between them. The choice of morphisms can determine the entire theory. Take a collection of mathematical objects. Different notions of map may produce different categories, different invariants, different equivalence relations, and different notions of classification. Thus even after an AI invents the right objects, another search problem remains:

What counts as a structure-preserving transformation?

The answer may not be obvious. Too restrictive a notion of morphism makes the theory brittle. Too permissive a notion destroys information. The useful choice is often the one under which the intended structure becomes

compositional. A future AI could discover morphisms simultaneously with objects. It might notice that certain transformations recur across examples. Those transformations form stable families under composition. Identity transformations arise naturally. An emergent category appears. The AI did not begin by deciding to use category theory. Categorical structure appeared because composition provided compression. This illustrates a larger possibility. Advanced AI need not be explicitly programmed to reproduce every abstraction humans know.

Some may emerge because they are efficient solutions to recurring representational problems. Others may not.

Mathematics Before Formalization

Formal mathematics is enormously important to machine reasoning because formal systems make correctness mechanically checkable. But formalization also introduces a danger. It is tempting to identify mathematics with whatever has been formalized. Human mathematical reasoning is richer. Mathematicians use diagrams. Metaphors. Geometric intuition. Approximate reasoning. Examples. Physical analogy. Dimensional arguments. Computational experiments. Informal pattern recognition. Partial definitions. Statements believed but not yet precisely formulated. Research lives for long periods in this ambiguous region. An AI restricted to already formalized objects may therefore miss an important part of mathematical creativity. Future systems may need two complementary modes. One is exploratory. Representations can be provisional. Definitions can change.

Analogies can be weak. Ideas can be generated cheaply. The other is rigorous. Claims are formalized. Assumptions are explicit. Proof obligations are tracked. Transformations are verified. Counterexamples are sought. The exploratory system generates possibilities. The rigorous system prevents fantasy from quietly becoming theorem. The interaction between the two may be more important than either alone. Too much rigor too early can suppress exploration. Too little rigor allows error to accumulate. Human mathematicians already move between these modes. Advanced AI could make the transition explicit and systematic.

Conjecture Generation

Theorem proving asks the machine to traverse a search space from premises to conclusion. Conjecture generation asks it to decide which conclusions are worth attempting. One strategy is empirical. Generate examples. Compute quantities. Look for regularities. If an invariant appears constant across millions of cases, conjecture that it is always constant. If a sequence matches

a known form, search for structural reasons. If an inequality repeatedly approaches the same bound, conjecture extremality. But mere pattern detection is not enough. Many numerical patterns are accidental. A strong system would need to distinguish fragile regularities from structural ones. It could perturb examples. Change dimensions. Alter boundary conditions.

Randomize coordinates. Replace commutative operations with noncommutative ones. Test neighboring axiom systems. A pattern surviving many transformations is more likely to reflect structure. This suggests that conjecture generation and invariance detection may become closely linked. The AI asks not only:

What appears true?

but:

Under what transformations does it remain true?

That is already movement toward deeper mathematics.

The Search for Generality

One of the characteristic moves of mathematics is generalization. A theorem holds for triangles. Does it hold for polygons? A statement holds over real numbers. What about complex numbers? A finite-dimensional result suggests an infinite-dimensional version. A property of groups suggests an analogue for categories. Generalization can expose the true structure behind an apparently special result. But indiscriminate generalization is useless. Every theorem can be generalized trivially by restating it in an artificially large framework. The valuable generalization identifies exactly which assumptions matter. An AI could search this space systematically. Start with a proven theorem. Remove one hypothesis. Search for counterexamples. If none appear, attempt proof.

Replace a structure with a weaker one. Repeat. Vary the ambient category. Change the coefficient domain. Increase dimension. Relax regularity. Alter finiteness conditions. The system constructs a lattice of possible statements surrounding the original theorem. At some boundary, truth fails. The shape of that boundary reveals the theorem's natural level of generality. This could automate something mathematicians normally discover slowly. The final result would not merely say that a theorem is true. It would explain **how true it can possibly be**.

The Search for Specialization

The opposite direction is equally important. Sometimes general theories are too broad to reveal the interesting phenomenon. A powerful AI could discover exceptional subclasses where additional structure appears. Among all groups, which subclasses admit unusually strong classification? Among all dynamical systems, where do new invariants arise? Among all manifolds, what additional hypotheses make a difficult problem decidable? Special cases are not necessarily failures of generality. They can be mathematical ecosystems of their own. An AI exploring enormous structure spaces could detect clusters in which unusually many theorems become available. It might then define the cluster. A new mathematical class is born. Again, the act of discovery is conceptual rather than deductive.

Proof as Certification

If these developments occur, the role of proof may change without becoming less important. Proof would remain the final arbiter of deductive validity. But it would increasingly function as **certification within a much larger discovery pipeline**. An artificial system might: generate examples, detect a pattern, invent a representation, formulate a conjecture, search for counterexamples, modify the hypotheses, discover a stronger abstraction, identify the correct transformations, derive secondary conjectures, and only then produce formal proofs. Proof certifies the result. It does not exhaust the intelligence responsible for finding it. This distinction matters when evaluating mathematical AI. A system that proves a difficult theorem is impressive. A system that identifies the theorem nobody knew needed to exist may be more significant.

A system that invents the concept under which an entire family of theorems becomes trivial would be more significant still.

From Solving Problems to Reorganizing Mathematics

Imagine two future AI systems. The first is an extraordinary theorem prover. Give it a formally stated conjecture and it finds a proof whenever one is computationally accessible. The second is slower at raw proof search. But it continuously analyzes mathematical knowledge. It discovers that several theories share the same hidden pattern. It invents a new abstraction. It restructures thousands of results around the abstraction. Theorems that once required separate proofs become instances of one general mechanism. Which system is the greater mathematician? Human mathematical history strongly suggests the second. The mathematicians remembered for transforming fields

frequently did so by reorganizing knowledge rather than simply by solving the largest number of problems.

The creation of a new representation can make entire generations of previous problems disappear. Artificial intelligence may eventually operate at this level. It may look at our mathematics not as a collection of problems waiting for solutions but as an inherited architecture waiting for reconstruction. It could ask: Why are these concepts separate? Why is this hypothesis repeated? Why do these proofs share the same shape? Why are these objects represented differently? What new definition eliminates this redundancy? What transformation exposes the invariant? What theory would make all of this inevitable? These are not proof questions. They are questions about the organization of mathematical thought.

And they point toward a future in which the machine mathematician does not merely derive consequences from our mathematics. It redesigns the conceptual landscape in which consequences are derived. That possibility makes the next question unavoidable. If mathematical difficulty depends partly on representation, then perhaps the most powerful capability an artificial mathematician could develop is not proof search at all. It is the ability to move from one mathematical world to another until the problem changes form. That is the subject of the next chapter.

CHAPTER 3

The Discovery of the Right Representation

Many mathematical problems are difficult only because they are being viewed in the wrong language. This sounds almost paradoxical. A problem appears to be an intrinsic object. Either it is hard or it is not. But mathematical practice repeatedly shows otherwise. An equation may resist direct manipulation yet become trivial after a substitution. A geometric construction may be opaque until translated into coordinates. A matrix problem may simplify after diagonalization. A nonlinear system may reveal hidden structure through symmetry. A topological question may become algebraic. An algebraic question may become geometric. A combinatorial counting problem may collapse once encoded by a generating function. A theorem that is tedious in one formalism may become almost automatic in another.

The object has not changed. The representation has. That observation suggests a profound role for advanced artificial intelligence in mathematics. Perhaps one of the most important things a machine mathematician could learn is not how to solve problems in the representations humans give it, but how to **search for representations in which the problems become easier**. This would amount to a new kind of mathematical computation. Not computation over numbers. Not computation over symbolic expressions. Not even computation over proofs. It would be computation over **ways of seeing**.

Representation Is Mathematical Leverage

A representation selects which features of an object are explicit and which remain hidden. Coordinates reveal some relations and obscure others. A graph emphasizes connectivity. A matrix exposes linear action. A group emphasizes symmetry. A category emphasizes transformation. A topological space emphasizes continuity. A metric space emphasizes distance. A probability distribution emphasizes uncertainty. A type emphasizes permissible construction. Each representation changes the operations that become natural. This matters because mathematical reasoning is partly a matter of choosing operations that align with the structure of the problem. If the representation is poorly matched, every step feels artificial. If the representation is well matched, the same structure may almost solve itself.

A classic mathematical experience is the sudden disappearance of difficulty after the right change of viewpoint. One struggles with a problem for days. Then a substitution is noticed. A coordinate system is rotated. A conserved quantity appears. A quotient is taken. The object is factored through a universal construction. What seemed complicated becomes simple. The sensation is not that one has computed faster. It is that one has stopped asking the problem in the wrong language.

Humans Search Representations Informally

Mathematicians already perform representation search. They simply do not usually call it that. A researcher confronted with an unfamiliar problem may try drawing the object. Then writing equations. Then looking at examples. Then computing invariants. Then introducing coordinates. Then eliminating coordinates. Then passing to a quotient. Then lifting to a covering space. Then embedding the problem in a richer structure. Then restricting it to a simpler one. Each move changes the representational environment. The search is guided by experience, analogy, taste, and intuition. A mathematician recognizes that a problem “looks like” something encountered before. The analogy may be weak. Perhaps only one diagram resembles another.

Perhaps an equation has the same symmetry. Perhaps a degeneration occurs in a familiar way. The mathematician experiments. Most such transformations lead nowhere. A few reveal structure. The process is extraordinarily powerful but also limited. A human can consider only a small number of representations seriously. An artificial intelligence could potentially search thousands or millions. That quantitative change might eventually become qualitative.

The Space of Representations

Imagine that a mathematical problem is not given one representation but a family of possible encodings. The same underlying situation might be represented as: an equation, a system of constraints, a graph, a tensor, a geometric object, a transformation group, a category, a logical theory, a rewriting system, a dynamical process, a probabilistic model, a program, or something not yet named. Each representation provides different primitive operations. Each makes some invariants cheap to calculate. Each induces different notions of similarity. Each suggests different proof strategies. An advanced AI could construct a graph whose nodes are representations and whose edges are transformations between them. One edge might represent a Fourier transform.

Another a change of basis. Another a categorical equivalence. Another a dualization. Another a coordinate chart. Another a quotient. Another a functor. Another an encoding of a logical theory into algebra. Another a passage from a combinatorial object to a topological realization. The machine could then search this graph. The task would no longer be:

Find a proof of the theorem.

It would become:

Find a path through representation space that transforms the theorem into something easy to prove.

This resembles compilation. A high-level program is transformed into intermediate representations until it reaches a form suitable for execution. Future machine mathematics may similarly require many intermediate mathematical languages.

Mathematical Compilation

Suppose a theorem is posed in geometry. The AI notices a symmetry group. It translates the geometry into an action of that group. The action has a natural invariant ring. The ring is encoded algebraically. The algebra admits a decomposition. The decomposition yields a combinatorial object. The combinatorial statement is solved by a finite argument. The proof is then translated backward into the original geometric language. No single representation contained the whole solution. The solution was the path between representations. Human mathematicians sometimes construct arguments of exactly this kind. But AI could make such cross-representational search routine. One could imagine a future mathematical system maintaining libraries not merely of theorems but of **translation operators**.

For example: geometric object -> coordinate model, coordinate model -> polynomial constraints, polynomial constraints -> ideal, ideal -> combinatorial signature, combinatorial signature -> graph, graph -> homological invariant, invariant -> categorical object. The transformations would carry explicit proof obligations. The AI could compose them automatically. Mathematics would begin to resemble an ecosystem of interoperable representations.

Difficulty as Description Length

Representation search may connect naturally with compression. A good representation often shortens the description of a phenomenon. Suppose a system contains a thousand observed identities. Represented separately, the

data are complex. Then a hidden symmetry is discovered. The thousand identities become consequences of one group action. The group-theoretic description is shorter. This is not merely aesthetic. The shorter representation exposes causal or structural organization. An AI might therefore score candidate representations according to how much they compress known facts. If representation R1 requires one thousand independent statements while representation R2 explains them using ten principles, R2 has strong advantages. This does not prove that R2 is philosophically deeper.

But it is a measurable sign that the representation captures recurring structure. Compression may therefore become a guide to mathematical discovery. A machine might search for transformations that minimize something analogous to:

description of definitions + description of assumptions + description of proofs + description of exceptions.

A representation that drastically reduces this total becomes interesting. In that sense, a mathematical theory can be viewed as a compression algorithm for a family of mathematical facts.

Coordinate Systems as an Elementary Example

Coordinate choice illustrates the idea in a familiar setting. An object may be described relative to many coordinate systems. The underlying geometry is unchanged. But calculations can differ dramatically. A symmetric object represented in a badly chosen frame may require complicated equations. Choose coordinates aligned with the symmetry and terms vanish. A matrix may appear dense. Change basis and it becomes diagonal. The diagonal form makes repeated powers trivial. Nothing about the underlying linear transformation has changed. Only its representation has. Humans learn standard transformations because they repeatedly simplify important classes of problems. An AI could generalize this idea dramatically. Instead of selecting among a handful of standard coordinate systems, it could search huge families of transformations for those that minimize complexity.

The optimal transformation might have no standard name. It may not correspond to anything a human would naturally try. This raises a fascinating possibility. There may exist many mathematical problems currently regarded as deep because humans have not discovered their simplifying coordinates.

Representation Discovery Versus Representation Selection

Selecting a representation from a known library is one challenge. Inventing a new representation is much harder. An AI may eventually do both. Suppose existing encodings fail to simplify a family of problems. The machine notices recurring fragments in the unsuccessful representations. It creates a new intermediate object that preserves exactly those fragments while discarding irrelevant structure. This new object has its own operations. Certain transformations become natural. Several difficult problems simplify. A new representation has been invented. At this point representation discovery merges with concept formation. The AI has effectively introduced a new mathematical language. This is historically common. Complex numbers created a new representation for algebraic phenomena.

Vectors reorganized geometry and mechanics. Matrices provided a language for linear transformations. Tensor notation allowed coordinate-independent relationships to be manipulated systematically. Distributions extended the notion of function. Categories created a language for compositional relations among structures. Each innovation made previously awkward patterns manageable. An AI capable of inventing representations could perform analogous transitions without waiting for generations of human mathematical development.

The Danger of Arbitrary Representation

Not every transformation is helpful. With enough freedom, any finite collection of facts can be encoded in a representation that makes them appear trivial. One could build the answers into the encoding itself. This would provide compression only superficially. A useful representation must preserve more than the data used to construct it. It should generalize. It should expose structure that predicts new facts. It should remain stable under meaningful transformations. It should support composition. It should connect with other domains. It should not merely memorize. This distinction will be crucial for machine mathematics. A powerful AI could generate representations so flexible that apparent breakthroughs are simply disguised overfitting.

Mathematical rigor therefore needs to extend beyond proof correctness. The system must track whether the representation itself contains hidden assumptions or problem-specific information. A representation discovered from a dataset of examples should be tested on unseen examples. A proposed invariant should survive perturbations. A translation should be invertible

when equivalence is claimed. A simplification should not quietly discard exceptional cases. The machine must be able to distinguish genuine structural compression from elaborate encoding tricks.

Invariants as Representation Anchors

Invariants provide one safeguard. When different representations describe the same underlying object, certain properties should persist. An AI moving through representation space could track these preserved quantities. Suppose an object is transformed through several encodings. At each stage the representation changes dramatically. If a claimed equivalence is valid, appropriate invariants should agree. This creates a form of mathematical checksum. The precise invariants depend on context. Dimension. Rank. Homology. Spectrum. Cardinality. Euler characteristic. Group structure. Logical satisfiability. Information measures. Or machine-discovered invariants. The more important point is methodological. Representation search should not be free-form symbolic mutation. It should be a controlled process in which transformations carry explicit guarantees about what is preserved.

This could make machine mathematics simultaneously more creative and more rigorous.

Equivalence as a Search Target

A sufficiently advanced system might organize much of representation search around equivalence. Two structures may look different but encode the same mathematical information. Finding an equivalence can turn one difficult theory into another familiar one. Sometimes the relationship is weaker than equivalence. There may be an adjunction. A duality. A faithful embedding. A quotient. A covering. A relaxation. A completion. Each relation creates another possible route through mathematical space. The AI need not insist that every transformation preserve everything. It may deliberately discard information when the discarded information is irrelevant to the problem. This is already common in mathematics. To solve a problem, one often passes to an invariant that forgets most details while preserving the feature of interest.

An effective representation system therefore needs a hierarchy of transformation strengths. Some transformations preserve exact identity. Some preserve equivalence. Some preserve only a selected invariant. Some produce approximations. Some create bounds. A mature machine mathematician would know which level is sufficient for the task.

Representation and Proof Complexity

There is a direct relationship between representation and proof search. Formal theorem proving can become computationally expensive because the search space is enormous. But a change of representation can restructure that space. A theorem requiring thousands of low-level deductions may become a one-line consequence of a higher-level lemma after the correct abstraction is introduced. Thus improvement in automated theorem proving may eventually depend less on faster search within a fixed logic and more on changing the vocabulary available to the prover. If proof search repeatedly encounters the same pattern, the machine could invent a lemma. If families of lemmas repeatedly share structure, it could invent an abstraction.

If the abstraction produces cumbersome arguments, it could search for an alternative representation. Proof optimization becomes representation optimization. This resembles how mathematical expertise develops in humans. Experts do not merely search more quickly through elementary operations. They perceive larger chunks. A novice sees many steps. An expert sees one structure. Advanced AI might automate the creation of those chunks.

The Library That Rewrites Itself

Traditional mathematical libraries grow by accumulation. New definitions are added. New theorems are proved. New files appear. A future AI-maintained library could periodically reorganize itself. Suppose thousands of results have accumulated around several related definitions. The AI discovers a more general structure that subsumes them. It introduces the new concept. Existing theorems are refactored. Redundant lemmas disappear. Proofs shorten. Translations to old terminology are preserved for compatibility. The library has not merely become larger. It has become conceptually better organized. This would be analogous to refactoring in software engineering. But the object being refactored is mathematics itself. A self-refactoring mathematical corpus could continuously seek a smaller and more coherent conceptual basis.

Such a system would make mathematical knowledge dynamic in a way printed literature is not.

The Problem of Human Readability

Representation search leads quickly to a difficult question. What if the most effective representation is incomprehensible to humans? Suppose an AI discovers a mathematical encoding in which a major unsolved problem becomes easy. The internal representation involves a structure with thousands

of interacting components. The machine provides a formally verified proof. But the representation has no intuitive human interpretation. Has the problem been understood? In one sense, yes. The theorem is proved. In another sense, perhaps not. Humans often demand more than correctness. We want conceptual access. We want to know what is happening. This suggests that future mathematical AI may need two optimization targets:

machine efficiency and **human interpretability**. The best native representation for the machine may differ from the best explanatory representation for us. The system could therefore search for translation layers. First solve the problem in whatever representation works. Then compress the solution into a form that preserves as much of the underlying insight as possible while remaining cognitively manageable. This would make explanation itself a representational problem.

Machine-Native Coordinates

Humans developed notation adapted to human sensory and cognitive systems. We favor two-dimensional diagrams because we see spatially. We prefer short symbolic expressions because working memory is limited. We arrange proofs sequentially because language is sequential. An AI need not share these constraints. Its natural representation of a mathematical object might be distributed across a high-dimensional latent structure. Relationships that require pages of human notation might be locally adjacent in the machine's internal space. What we experience as a complicated chain of transformations might be a single learned operation. This should not automatically be dismissed as opaque numerical computation. If the internal structure supports exact operations, formal verification, compositionality, and reusable abstraction, it may constitute a legitimate mathematical language.

The challenge would be externalization. Human mathematics may eventually require something analogous to scientific visualization for machine-discovered abstractions. We may need projections. Interactive interfaces. Simplified analogues. Lower-dimensional shadows. Approximate narratives. Different users may need different explanations of the same machine-native object. The theorem remains one thing. Its human representations multiply.

A Mathematics of Translation

If machine-native representations become common, translation may itself emerge as a major mathematical discipline. One theory might optimize proof. Another explanation. Another computation. Another visualization. Another physical interpretation. The relationships among them become objects worthy

of study. Instead of asking which representation is the true one, mathematics may increasingly ask which transformations preserve which structures. This is already a familiar mathematical instinct. Advanced AI could elevate it into a general architecture of mathematical knowledge. Every result could carry not only a proof but a map of its representations. The system might know: where the theorem is easiest to prove, where it is easiest to compute,

where its geometric meaning is clearest, where its generalization is most natural, and where a human is most likely to understand it. Mathematical knowledge would become navigable rather than merely stored.

Representation Search as Intelligence

This returns us to a central question. What does mathematical intelligence consist of? If the answer is merely deduction, then faster theorem proving may approximate it well. But mathematical practice suggests something richer. Intelligence includes knowing what to ignore. What to preserve. Which analogy matters. When to change coordinates. When to introduce a new object. When two definitions are secretly describing the same phenomenon. When a complicated expression should be replaced by an invariant. When the whole problem should be moved into another domain. These are representational judgments. A machine capable of making them systematically could possess a kind of mathematical flexibility difficult to achieve through proof search alone.

It might confront a theorem and decide not to prove it. Instead it would transform the objects. Reframe the question. Invent a new representation. Discover an equivalence. And return with a proof of a statement that suddenly makes the original problem obvious. That would feel much closer to mathematical creativity.

The First Major Break

The first unmistakable sign of machine-born mathematics may therefore not be a theorem humans failed to prove. It may be a representation humans failed to imagine. The AI might introduce an unfamiliar mathematical object and say: These thirty problems become instances of the same construction here. At first the new representation could appear unnatural. Humans may resist it. Then the proofs shorten. Unexpected connections emerge. Computations simplify. New conjectures become visible. Students eventually learn the representation routinely. What once looked alien becomes obvious. That is how successful mathematical abstractions have often entered human culture. The difference would be the source. For the first time, the new mathematical viewpoint would not have originated in a human mind.

And once machines can search effectively for representations, another possibility follows almost immediately. If representations are searchable, then so are the concepts from which those representations are built. Definitions themselves cease to be fixed inputs. They become candidates. Mathematics begins searching not only for answers but for the language in which answers can exist. That is where machine mathematical discovery becomes considerably more radical.

PART II
When Mathematics Becomes Searchable

CHAPTER 4

Searching the Space of Definitions

A theorem can be difficult because the proof is hard. It can also be difficult because the definition is wrong. Not logically wrong. Definitions are not ordinarily true or false. But a definition can be badly chosen. It can conceal the important structure, include pathological cases, exclude natural examples, behave poorly under composition, or force every theorem into an awkward form. Mathematics has repeatedly advanced by improving its definitions. This suggests a striking possibility for artificial intelligence:

definitions themselves may become objects of search.

Instead of treating mathematical concepts as fixed human inputs, an AI could generate, test, compare, refine, and discard candidate definitions. That would move machine mathematics into a much deeper creative regime. The machine would no longer ask only:

What follows from this concept?

It could ask:

What concept should exist here at all?

Definitions as Compression Devices

A useful mathematical definition does more than name a collection of objects. It compresses structure. Consider a family of examples that share many properties. One could describe each example independently. Or one could identify the common feature responsible for the shared behavior and define a new class around it. The definition replaces repetition. A powerful concept often causes many theorems to reorganize themselves around a common core. This gives a practical criterion for machine-generated definitions. A good definition should reduce the total descriptive complexity of a body of mathematics. Suppose an AI observes one thousand structures satisfying a collection of recurring properties. It proposes a definition D.

Under D, five hundred separate results collapse into twenty general theorems. This is evidence that D has captured something real. Not metaphysically real in some simple sense, but structurally useful. The definition has found a recurring mathematical regularity. An AI could measure this effect. Candidate definitions could be scored according to: how many examples they unify, how many theorems they compress, how naturally they interact with known constructions, how stable they remain under

transformation, how few arbitrary clauses they require, and how much genuinely new mathematics they generate. Definitions become hypotheses about structure.

The Search Space Is Enormous

The challenge is that the space of possible definitions is effectively unbounded. Even over a small mathematical vocabulary, one can combine conditions endlessly. Require associativity. Drop commutativity. Add an identity. Demand invertibility. Impose continuity. Require a universal property. Restrict dimension. Add a topology. Remove compactness. Require preservation under a family of maps. And so on. Most combinations will be mathematically uninteresting. An AI therefore needs a way to navigate this space intelligently. One strategy is to begin from examples. Given a set of structures that appear related, the system searches for predicates that separate them from nearby counterexamples. This resembles concept learning. But mathematical concept learning is more demanding than ordinary classification.

A good definition should not merely classify correctly. It should generate theory.

Classification Is Not Enough

Suppose an AI is given two groups of objects and finds a bizarre numerical condition that perfectly distinguishes them. Perhaps every positive example happens to satisfy a large polynomial identity involving arbitrary encodings. As a classifier, the rule is excellent. As mathematics, it may be useless. The rule explains nothing. It does not compose. It does not expose invariants. It does not connect naturally to other structures. It may fail immediately outside the training set. A mathematically good definition has stronger properties. It should help answer new questions. It should suggest natural morphisms. It should survive translation between representations. It should admit examples and counterexamples for understandable reasons.

It should sit productively inside a larger conceptual network. Thus the machine must distinguish **predictive concepts** from **structural concepts**. The difference may prove central to mathematical AGI.

Definition by Closure

One powerful source of definitions comes from closure. Suppose a family of objects behaves well under certain operations. Whenever two such objects are combined, the result remains in the family. Subobjects remain in the family.

Limits preserve the family. Products preserve the family. Duality preserves the family. An AI may notice these regularities before it understands why they occur. It could then search for the weakest definition explaining the closure properties. This reverses the usual direction of textbook presentation. Humans often state the definition first and later prove closure theorems. The AI might infer the definition from the closure behavior. For example: observed structure -> closure pattern -> candidate invariant -> definition.

The result could be a concept whose motivation is computational rather than historical.

Definition by Invariance

Another route is invariance. A mathematical property becomes particularly interesting when it survives transformations considered irrelevant to the underlying structure. If a quantity changes under a coordinate transformation, it may depend on representation. If it survives, it may reflect something deeper. An AI could search for properties that remain constant under families of transformations. Those properties become candidate invariants. Families of objects sharing an invariant may then suggest new definitions. This creates a hierarchy: transformations define equivalence, equivalence suggests invariants, invariants suggest classes, classes suggest theories. The system could discover these layers jointly. It may not need humans to tell it first what the objects are, then what the morphisms are, then what the invariants should be.

The whole package could emerge from relational structure.

Definition by Failure

Counterexamples provide another route. Suppose a proposed theorem fails. Instead of abandoning the theorem, the AI analyzes the counterexample. What property distinguishes the counterexample from the successful cases? Perhaps the failure occurs only when the space is disconnected. Or noncompact. Or singular. Or infinite-dimensional. Or noncommutative. The machine adds a condition. The theorem now holds across all tested cases. But another counterexample appears. A second refinement follows. This is one of the oldest patterns in mathematical development. AI could automate it aggressively. The process becomes: conjecture -> counterexample -> diagnosis -> revised definition -> new theorem. Over time, entire concepts could crystallize from the repeated need to exclude particular failure modes.

A mature mathematical theory might emerge not from inspiration but from systematic adversarial repair.

The Minimal Definition

There is an important optimization problem here. Suppose several different definitions make the desired theorem true. Which should the AI prefer? Usually the weakest sufficient definition is attractive. If a theorem requires only assumptions A and B, adding C, D, and E is unnecessary. But minimality alone is not enough. The weakest possible condition can be ugly and unstable. A slightly stronger definition may produce a much richer theory. This creates a tradeoff among: generality, simplicity, closure, computability, explanatory power, and compatibility with other structures. Human mathematicians negotiate these tradeoffs by taste. A machine could make them explicit. One could imagine a Pareto frontier of definitions.

No single candidate is best in every sense. One definition is maximally general. Another gives the cleanest morphisms. Another minimizes proof complexity. Another produces the strongest classification theorem. Another connects best to existing theory. The AI might retain several. Mathematical ontology could become plural rather than singular.

Competing Definitions

This already happens in human mathematics. The same intuitive phenomenon may admit several definitions that coincide under common hypotheses but diverge in edge cases. Different branches of mathematics may favor different versions. An AGI could explore these choices systematically. It could ask: Where do definitions D1 and D2 agree? Where do they differ? Which theorems survive under each? Which version composes better? Which admits stronger computational methods? Which corresponds more naturally to geometric intuition? Which has better categorical properties? Instead of arguing vaguely over the “right” definition, the system could map the entire definition landscape. The output might be a diagram showing several notions connected by implication, equivalence under added hypotheses, and counterexample boundaries.

This would turn definitional disputes into structured mathematical objects.

Definitions as Programs

A definition can also be viewed operationally. To decide whether an object belongs to a class, one evaluates a predicate. Some predicates are computationally easy. Others are undecidable. Others require expensive global reasoning. This introduces a computational dimension to concept design. Suppose two definitions are mathematically equivalent but one admits efficient recognition and the other does not. A machine mathematician may

strongly prefer the computationally tractable formulation. Or it may maintain both: one for conceptual clarity, another for computation. This mirrors the distinction between declarative and algorithmic mathematics. The AI could search explicitly for computationally favorable characterizations of abstract concepts. A structure originally defined by existence might later receive an equivalent constructive characterization.

A global property might acquire a local test. An exponential procedure might collapse into polynomial time after the right invariant is found. In this way, concept discovery and algorithm discovery become intertwined.

Discovering the Right Level of Abstraction

Definitions can fail by being too concrete. Suppose many theorems concern matrices. An AI notices that the actual statements depend only on the linear transformations represented by those matrices. The coordinates are irrelevant. It replaces matrices with linear maps. Later it notices that many arguments depend only on composition and identity. It abstracts further. The pattern becomes categorical. But abstraction can also go too far. A theory may become so general that its strongest results disappear. The machine therefore needs to discover the productive level of abstraction. This is not simply "the most general theory." The correct level is where meaningful regularities remain visible. Too concrete: duplication.

Too abstract: emptiness. A powerful AI could search this scale. For each proposed abstraction, it could measure what is gained and lost. Which theorems unify? Which invariants disappear? Which algorithms become unavailable? Which new morphisms appear? The result could be an automated theory of abstraction depth.

Mathematical Species Formation

One way to visualize this process is biological. Concrete mathematical objects form populations. The AI computes properties and relations among them. Clusters appear. Some clusters are unstable. Others persist under many representations. The system proposes definitions around stable clusters. Those definitions generate new populations of examples. Further substructure appears. Some concepts split into several more natural subclasses. Others merge. Entire taxonomies evolve. Mathematical concepts begin to resemble species formed by structural selection. The analogy should not be pushed too far, but it captures an important idea. A machine-generated mathematical ontology need not be designed once. It could evolve continually as new examples and theories appear.

Automatically Invented Morphisms

Definitions of objects alone are rarely sufficient. Once a new class has been identified, the AI must ask how objects in the class relate. This may be harder than inventing the class itself. Consider two candidate notions of map. The first preserves every visible feature. It is too rigid. Almost no interesting maps exist. The second preserves almost nothing. It is too loose. The theory loses structure. The right morphisms often preserve exactly the information that matters. An AI could search for them by examining transformations occurring naturally across examples. Suppose certain transformations repeatedly compose. Identity transformations exist. Important invariants are preserved. The transformations interact well with other constructions.

This strongly suggests a useful morphism class. Then a category may emerge. The machine has not been told to create a category. It has discovered compositional structure because composition compresses behavior. This may be one route by which advanced AI rediscovers familiar abstract mathematics independently.

And What If It Does Not Rediscover It?

This raises a more provocative question. Suppose an AI is allowed to develop mathematical concepts freely. Must it rediscover groups? Categories? Topoi? Homotopy types? Perhaps. These structures recur so widely that a powerful independent mathematical intelligence may encounter them naturally. But perhaps not in the form humans recognize. The machine may discover a more general primitive from which several human concepts appear as special projections. From its perspective, our separate theories may look like different coordinate patches on one larger structure. Alternatively, it may choose a completely different organization because its computational architecture favors different primitives. This is where machine mathematics could become truly alien.

The issue would not be that the AI knows concepts humans do not. Humans have always discovered new concepts. The difference would be that the machine's *ontology of mathematics* could diverge from ours.

Theory Generation from a Definition

Once a useful definition is found, an AI need not stop. It can ask automatically: What are the simplest examples? What are the exceptional examples? What are the automorphisms? What are the natural maps? What subobjects exist? What quotients exist? What products? What duals? What invariants? What classification problems arise? What is the deformation

theory? What happens in low dimension? What stabilizes in high dimension? Which existing theories embed? Which new conjectures appear? This is essentially the automatic generation of a research program. A single definition could produce thousands of mathematically meaningful questions. The machine could prioritize them according to explanatory potential. Thus a future AI might not merely invent isolated concepts.

It could create entire fields.

Barren and Fertile Mathematics

Most possible definitions will lead nowhere. A candidate structure may admit few examples. Or too many. Its invariants may all be trivial. Its classification problem may simply reproduce another theory. Its morphisms may not compose interestingly. Its theorems may consist largely of restatements. The AI needs to identify fertile regions. One possible measure is theorem density. Another is compression. Another is connectivity to other theories. Another is diversity of nontrivial examples. Another is existence of rich symmetry. One could imagine the machine assigning something like a fertility score to a newly generated theory. High-fertility theories get more computation. Low-fertility theories are archived. This would create evolutionary pressure within mathematical exploration.

The Risk of Mathematical Junk

A system capable of generating definitions at enormous scale could produce an overwhelming amount of mathematically correct but useless material. This may become one of the central problems of machine mathematics. The bottleneck shifts from generation to selection. A machine can generate a million new definitions. Which ten matter? Human mathematical culture currently performs this selection slowly through peer attention, elegance, applicability, depth, and historical development. Machine mathematics will need stronger filters. Otherwise automated discovery could drown mathematics in technically novel debris. The distinction between novelty and importance becomes crucial. A structure is not interesting merely because nobody has written it down before. A theorem is not profound merely because it is new.

The real achievement lies in reorganizing understanding.

Conceptual Compression as Selection

Compression offers one selection principle. A newly invented concept is valuable if it substantially reduces the complexity of existing knowledge or

generates unexpectedly powerful consequences. Suppose concept A produces fifteen new theorems but connects to nothing else. Concept B unifies three previously separate theories, shortens hundreds of proofs, and predicts a new family of invariants. Concept B has greater conceptual leverage. The AI can measure this. This does not eliminate mathematical taste. But it begins to operationalize part of it. Perhaps future mathematics will develop explicit metrics for conceptual power. A definition may be evaluated by how much mathematical structure it explains per unit of descriptive complexity.

The Machine Says: This Concept Is Missing

Imagine a mature AI examining a large mathematical corpus. It detects recurring inefficiency. The same hypothesis appears in many unrelated-looking papers. The same diagram is proved commutative repeatedly. Several invariants have nearly identical definitions. Certain constructions repeatedly require ad hoc patches. The AI proposes a new structure. After introducing it, all of these repetitions vanish. It then proves a general theorem from which dozens of known results follow. This may be one of the most important signatures of genuine machine mathematical creativity. Not:

> **I proved your conjecture.**
>
> But:
>
> **Your mathematics is missing a definition.**

The New Unit of Discovery

Historically, mathematical achievement is often measured in theorems. Future machine mathematics may force a broader vocabulary. The important discovery could be: a definition, a representation, a morphism class, an invariant, an equivalence, a translation, a proof architecture, a new foundational layer, or an entire theory. The theorem becomes only one unit among many. This changes how mathematical progress itself might be measured. The most powerful machine may not be the one producing the largest number of proofs. It may be the one that most effectively reduces the conceptual complexity of mathematics while expanding its explanatory reach. That is a very different objective. And it leads naturally to another essential component of autonomous mathematical development.

If AI is going to invent definitions, it must become extraordinarily good at destroying bad ones. It needs not only creativity. It needs adversaries. It needs pathological examples. It needs edge cases. It needs systematic failure. In short, it needs counterexamples not as embarrassments, but as one of the primary engines of mathematical discovery.

CHAPTER 5

Counterexamples as a Creative Engine

Mathematics is often narrated as a sequence of successful proofs. A conjecture is proposed. A theorem is proved. A theory advances. The failures are usually compressed into footnotes, historical remarks, or omitted entirely. But failure is one of the principal engines of mathematical progress. A false conjecture can be more informative than a true one. A pathological example can reveal the hidden assumptions on which an entire theory depends. A single counterexample can force the invention of a better definition, a sharper invariant, or a new distinction that did not previously exist. For artificial intelligence, this matters enormously. A sufficiently advanced mathematical AI should not merely search for proofs.

It should search aggressively for ways to break its own ideas. That adversarial capacity may be as important as theorem proving itself.

The Productive Role of Failure

Suppose a machine generates the conjecture: Every object satisfying conditions A and B also satisfies property C. The most obvious next step is to search for a proof. But a better strategy may be to attack the statement first. Construct small examples. Vary parameters. Search degenerate cases. Relax hidden assumptions. Push the object to the boundary of its definition. Try high dimensions. Try low dimensions. Replace finite with infinite. Replace commutative with noncommutative. Replace smooth with singular. Replace connected with disconnected. If a counterexample appears quickly, enormous proof-search effort has been saved. But the counterexample offers something more valuable than efficiency. It tells the system where its understanding is incomplete.

Why did the conjecture fail? What feature distinguishes the failure from the successful cases? That question can produce new mathematics.

Counterexample as Diagnosis

A weak system treats a counterexample as rejection. A stronger system treats it as information. Suppose a theorem fails for one object X. The AI compares X with nearby examples where the theorem holds. Perhaps X is the only disconnected example. Or the only one with a singular point. Or the only one

lacking compactness. Or the only example with torsion. Or the only example whose symmetry group is nontrivial. The system identifies candidate causes. Then it tests them. Add connectedness as a hypothesis. Does the theorem now hold? If not, find another counterexample. Add compactness. Test again. This process can reveal the minimal structural conditions under which the statement becomes true.

The result is not merely a repaired theorem. It is a map of the theorem's boundary.

The Boundary of Truth

A mathematical statement rarely exists in isolation. Around it lies a family of nearby statements. Some are true. Some are false. Some become true only under additional hypotheses. Some fail only in exceptional cases. An AI could systematically explore this neighborhood. Take a theorem T with hypotheses H1, H2, H3. Remove H1. Search for a counterexample. Remove H2. Search again. Weaken H3. Change the ambient domain. Vary dimension. Replace equality with inequality. Replace exact structure with approximate structure. The result is a lattice of theorem variants. The boundary separating truth from falsity is itself mathematically interesting. It reveals which assumptions carry real structural weight. Human mathematicians often discover such boundaries over decades.

AI could attempt to map them systematically.

Minimal Counterexamples

Not all counterexamples are equally informative. A huge, pathological construction may prove that a statement is false but offer little intuition. A small counterexample can be devastatingly clear. The smallest group violating a conjecture. The lowest-dimensional manifold where a phenomenon fails. The shortest sequence breaking an inequality. The minimal graph defeating a proposed coloring rule. Minimal counterexamples isolate the essential obstruction. An AI could optimize explicitly for them. Whenever a conjecture fails, search for: the smallest cardinality, the lowest dimension, the minimal number of generators, the shortest description, the weakest violation, or the simplest structural obstruction. This makes failure interpretable. It also produces excellent mathematical explanations.

Pathology Libraries

Human mathematics has accumulated famous pathological examples. Functions that are continuous but nowhere differentiable. Spaces with

counterintuitive separation behavior. Groups with unexpected subgroup structure. Measures that violate naive intuition. Logical models that satisfy surprising theories. Such examples matter because they inoculate intuition against overgeneralization. A machine mathematician could maintain enormous automated pathology libraries. For every major concept, it could store objects that violate common intuitions. When a new conjecture appears, these examples are tested immediately. The system could even synthesize new pathologies tailored to the statement under consideration. Instead of waiting for a human expert to remember the right exotic example, the machine generates one.

Adversarial Mathematics

This suggests an architecture with competing agents. One agent proposes conjectures. Another searches for proofs. Another searches for counterexamples. Another weakens hypotheses. Another strengthens conclusions. Another searches for equivalent reformulations. Another checks whether a result is already implicit in known theory. The agents compete. A conjecture survives only after adversarial pressure. This resembles security testing. A system is not trusted merely because it works under normal conditions. It is attacked deliberately. Mathematical ideas could be hardened in the same way. A theorem that survives millions of intelligently generated adversarial examples becomes a stronger candidate for formal proof. A definition that remains useful after repeated attacks may be structurally robust.

The Value of Almost-Counterexamples

Even failed attempts to construct counterexamples can be informative. Suppose the AI searches an enormous class of objects and cannot find a violation. That does not prove the conjecture. But it can reveal something about where a proof might lie. Perhaps every attempted counterexample is forced into a structure satisfying a particular invariant. The failed search itself exposes a constraint. This can suggest a lemma. Thus counterexample search and proof search are not opposites. They can feed one another. A proof may emerge from understanding why every attempted counterexample collapses.

Counterexample-Guided Definition Formation

Now consider definitions. An AI proposes a class of objects defined by property D. The class behaves badly. Products leave the class. Natural transformations are not preserved. An expected invariant becomes meaningless. Rather than abandoning the concept, the machine searches for

the smallest examples exhibiting failure. Those failures reveal what the definition omitted. The definition is revised. Then attacked again. Over time, a stable concept emerges. This process resembles counterexample-guided refinement in computer science, but applied to mathematical ontology. The loop becomes:

definition -> theorem -> counterexample -> diagnosis -> revised definition.

Concepts mature through failure.

Theorem Repair

Humans routinely repair false conjectures. A theorem may fail as stated but remain almost correct. An AI could automate this repair process. Given a counterexample, it might explore several options: add a missing hypothesis, weaken the conclusion, restrict to a subclass, replace equality by inequality, add an exceptional case, change the notion of equivalence, or reformulate the theorem in a more invariant language. The system could score repairs by simplicity and explanatory power. The best repair is not necessarily the one that preserves the original statement most closely. Sometimes the counterexample reveals that the original formulation was misguided. A more radical reformulation may expose the true phenomenon.

Discovering Obstructions

Many mathematical theories are organized around obstructions. A desired construction exists unless some invariant prevents it. A map extends unless an obstruction class is nonzero. A space admits a structure only when certain conditions vanish. A polynomial factors unless a particular condition fails. Counterexamples often point directly toward such obstructions. Suppose a machine observes that every failed case shares a common invariant O. It conjectures: The desired property holds exactly when O = 0. Now the counterexample has been transformed into structure. The failure is no longer accidental. It is measured. This is a particularly powerful mode of discovery because it converts pathology into classification.

From Counterexample to New Invariant

Imagine an AI exploring a family of objects. A conjecture holds for millions of examples and fails for a small set. The system compares the exceptional cases and discovers a numerical or algebraic feature common to them. It defines a new invariant. The invariant vanishes in the ordinary cases and becomes nonzero precisely where the theorem fails. A new concept has

emerged from the counterexamples. The system may then discover that the invariant appears in other problems as well. What began as an exception becomes a new mathematical object. This is one way machine mathematics could generate theories from failure.

Random Search Is Not Enough

Brute-force enumeration can find counterexamples in finite spaces. But advanced mathematics requires more. Many interesting counterexamples are extremely sparse. Random sampling may never encounter them. A sophisticated AI must search structurally. It needs models of how conjectures fail. If a theorem depends implicitly on compactness, search noncompact limits. If it depends on smoothness, introduce singularities. If it depends on finite generation, construct infinitely generated examples. If it depends on exact symmetry, perturb the symmetry. This requires understanding. Counterexample generation becomes itself a form of mathematical creativity. The system must imagine how the theorem might be wrong.

Machine Creativity Through Destruction

There is a common tendency to equate creativity with generation. Invent a theorem. Invent a definition. Invent a proof. But destruction can be equally creative. A good counterexample reveals an unexpected possibility. It shows that the conceptual landscape is larger than assumed. It breaks a false equivalence. It separates two notions thought identical. It exposes hidden structure. In this sense, mathematical creativity may alternate between construction and destruction. Build a concept. Break it. Repair it. Generalize it. Break the generalization. Discover the obstruction. Rebuild the theory around the obstruction. An autonomous mathematical system could run this cycle continuously.

Counterexamples and Mathematical Safety

There is another reason adversarial mathematics matters. Machine-generated mathematics can fail subtly. A theorem prover may be sound, but the formalization may encode the wrong statement. A symbolic transformation may introduce hidden assumptions. A numerical experiment may suggest a false pattern. A learned model may overgeneralize from training examples. A translation between representations may discard essential information. Counterexample search acts as a defense against these failures. Every transformation should invite adversarial testing. Every equivalence claim should be attacked. Every generalized theorem should be probed at its

boundaries. Every machine-discovered abstraction should be tested against synthetic pathologies. The objective is not merely mathematical productivity.

It is epistemic reliability.

The Counterexample Economy

Future mathematical AI may need an explicit economy of computational attention. Proof search can be expensive. Counterexample search can be cheap. Therefore the system might adopt a default policy: First try to kill the theorem. Only then invest heavily in proving it. This mirrors scientific falsification, though mathematics remains fundamentally different because proof ultimately determines deductive truth. Still, the computational logic is powerful. A false conjecture should ideally die early. A true conjecture should survive increasingly sophisticated attacks before formal proof begins. This creates a layered validation pipeline: cheap examples, structured examples, adversarial examples, boundary cases, formal model search, then proof. The system spends its strongest resources only on ideas that survive weaker tests.

Counterexamples Across Representations

A conjecture may appear plausible in one representation and obviously false in another. This connects adversarial search to representation search. Suppose a geometric theorem is difficult to test directly. Translate the objects into algebra. Now a small symbolic example reveals failure. Or transform a combinatorial conjecture into a graph problem where a counterexample can be enumerated. An AI capable of moving among representations gains a much stronger adversarial capacity. It can search for the domain in which falsification is easiest. The same principle applies to proof. Some representations are good for proving. Others are good for breaking. A mature system would know the difference.

Counterexample Networks

A single counterexample may invalidate many related conjectures. Future systems could organize counterexamples into networks. One object may violate five natural generalizations simultaneously. Another may distinguish two competing definitions. A third may expose a shared hidden assumption across several theories. The AI could identify especially informative counterexamples and treat them as canonical test objects. Mathematical development would then proceed against a permanent adversarial benchmark. Any new theorem or abstraction entering the system would be tested against

this evolving suite. This resembles regression testing in software engineering. Once a mathematical bug has been found, it should never silently return.

Learning From Historical Failure

A machine could also mine the history of mathematics for failed conjectures. Why did mathematicians believe them? Which examples were missing? Which intuitions misled them? Which new definitions repaired the situation? This produces a training corpus not of mathematical success, but of mathematical error. Such a corpus may be extraordinarily valuable. It teaches the machine not merely what is true, but **how intelligent reasoning goes wrong**. That knowledge could improve future conjecture generation. The AI might recognize patterns associated with historically dangerous overgeneralizations. For example: evidence only in low dimensions, overreliance on commutative intuition, assuming finite behavior extends to infinite cases, confusing local with global structure,

ignoring degenerate objects, or assuming coordinate expressions reflect invariant truth. A system aware of these failure modes could attack its own ideas more intelligently.

The Deep Role of Exception

Exceptions are sometimes treated as annoyances. But in mature mathematics, exceptional cases often contain the most structure. Exceptional Lie groups. Exceptional geometries. Exceptional dimensions. Singular fibers. Boundary phenomena. Degenerate cases. The exception may reveal that a classification theorem has a deeper architecture than expected. An AI exploring millions of structures could become exceptionally good at detecting statistically rare but mathematically rich regions. Instead of discarding outliers, it could ask whether they represent noise or new theory. That distinction is crucial. Machine learning often treats outliers as problematic data. Mathematics sometimes builds entire fields around them.

When the Counterexample Is Better Than the Theorem

Imagine an AI proposes a broad conjecture. It finds a counterexample. The counterexample turns out to possess a remarkable structure. Studying that structure leads to a new invariant. The invariant connects two previously unrelated fields. The original conjecture is forgotten. The failed idea produced something more important. This is entirely plausible. Research is not a linear optimization toward proving initial hypotheses. Unexpected failure can redirect the whole program. A genuinely autonomous

mathematical AI must be allowed to change its mind. It needs the ability to abandon goals when the failure itself becomes more interesting. That is a subtle form of agency in research.

Mathematical Falsification at Scale

The scale available to future AI could transform this process. Humans may test dozens of examples. A machine can test billions. Humans may know a few classic pathologies. A machine can synthesize vast families. Humans may inspect a handful of nearby definitions. A machine can explore whole lattices of alternatives. The consequence is not simply greater confidence. It could change the style of mathematics. Conjectures may become more sharply formulated before they reach human attention. Definitions may arrive already stress-tested. Boundary cases may be mapped systematically. Theorems may come with automatic descriptions of exactly how their assumptions can fail. This would make mathematical knowledge more robust.

Toward Experimental Mathematics

Once counterexample generation becomes large-scale and systematic, mathematics begins to acquire an experimental layer of unprecedented power. A machine constructs mathematical worlds. Measures them. Perturbs them. Destroys hypotheses. Records phase transitions. Searches for universal patterns. From these experiments emerge conjectures and concepts. Proof still matters. But proof arrives after a vast landscape has already been explored. This points toward the next transformation. If advanced AI can generate examples, attack conjectures, vary axioms, and map mathematical behavior across huge spaces, then mathematics begins to resemble something it has historically only partially been: an experimental science of abstract worlds. The laboratory contains no atoms. Its experiments take place in structures.

And the number of possible experiments is effectively unlimited.

CHAPTER 6

Experimental Mathematics at Machine Scale

Mathematics is usually distinguished from the empirical sciences by proof. Physics observes the world. Chemistry performs experiments. Biology measures living systems. Mathematics proves consequences from definitions and assumptions. The distinction is real, but incomplete. Mathematicians have always experimented. They calculate examples. Draw diagrams. Search numerical patterns. Test conjectures in special cases. Vary parameters. Run computer simulations. Enumerate finite structures. Look for invariants. Probe boundary cases. Observe what changes and what refuses to change. The difference is that mathematical experiments do not determine truth in the same way physical experiments do. A billion successful examples do not prove a universal theorem. But experiments can reveal what theorem should be proved.

With advanced artificial intelligence, this experimental side of mathematics could expand enormously. The scale would change first. Then the methodology. Eventually, perhaps, the structure of mathematical discovery itself.

A Laboratory of Abstract Worlds

A physical laboratory is constrained by matter, energy, time, cost, and measurement. A mathematical laboratory is constrained mainly by representation and computation. If a structure can be generated exactly, an AI can experiment on it. Finite groups. Graphs. Algebras. Codes. Cellular systems. Finite categories. Logical models. Combinatorial geometries. Dynamical systems. Discrete approximations to continuous structures. A machine can produce entire populations of such objects. Then it can measure them. Count automorphisms. Compute spectra. Determine homology. Search substructures. Test identities. Construct quotient objects. Compare invariants. Track behavior under transformations. This turns abstract structure into experimental material. An AI could run millions of such experiments without waiting for a human mathematician to formulate every query individually.

From Example to Population

Human mathematical intuition often develops from a small number of carefully chosen examples. This is powerful but dangerous. A handful of

examples may create false expectations. Low-dimensional cases can behave atypically. Small finite structures may hide phenomena appearing only later. Famous examples may be historically overrepresented. An AI can instead study populations. Suppose a conjectured property concerns a class of finite structures. The machine can enumerate every example up to a substantial size. It can determine how common the property is. Identify the first failure. Measure correlations with other invariants. Cluster the structures by behavior. Detect exceptional families. Track how frequencies change with size.

The result is not merely a list of examples. It is an empirical landscape.

Mathematical Phase Transitions

One especially interesting possibility is the discovery of phase-transition-like behavior in mathematical spaces. As parameters change, structural behavior may shift abruptly. Random graphs already provide familiar examples: connectivity and other properties can appear suddenly near threshold regimes. Similar phenomena may exist across many less-explored spaces. An AI could vary: dimension, density, number of generators, relation complexity, coefficient domains, topological constraints, symmetry assumptions, logical axioms, or combinatorial parameters. At certain thresholds, new structures may emerge. Classification may suddenly become difficult. Invariants may stabilize. Symmetry may break. Typical behavior may become exceptional behavior. These transitions could suggest entirely new mathematical questions.

Mapping Theory Space

Now enlarge the experiment. Do not vary only objects. Vary the axioms. Suppose a theory is defined by assumptions $A1,...,An$. An AI could systematically construct nearby theories by removing, weakening, strengthening, or replacing assumptions. Then ask: Which theorems survive? Which collapse? Which new models appear? Which invariants remain meaningful? Which theories become equivalent? Which become inconsistent? Instead of studying one theory, the machine studies a **space of theories**. Each point corresponds to an axiom set. Edges correspond to small modifications. The resulting map could reveal regions of mathematical stability. Some conclusions may persist across enormous families of axiom systems. Others may depend delicately on one assumption.

This would provide a new perspective on foundational structure.

The Experimental Discovery of Necessity

Traditional proof shows that a conclusion follows from hypotheses. Large-scale mathematical experimentation could help determine which hypotheses are actually responsible. Suppose a theorem is stated under six assumptions. An AI constructs models satisfying different subsets. It discovers that four assumptions are irrelevant. A fifth is needed only in one exceptional family. The sixth carries nearly all of the theorem's force. This does not replace proof. But it guides the machine toward a sharper theorem. Eventually the proof can certify the minimal statement. Experimental mathematics becomes a tool for discovering logical necessity before formalizing it.

Massive Enumeration

Enumeration is among the most direct forms of mathematical experimentation. For small finite spaces, one can generate every possible object. This produces exact empirical data. The difficulty is combinatorial explosion. Even simple structures can become astronomically numerous. AI may help not merely by increasing raw computation but by learning how to quotient the search space. Symmetry can eliminate redundant cases. Canonical forms can identify equivalent objects. Invariants can partition populations. Search can focus on representatives. The machine might learn increasingly sophisticated ways to enumerate only mathematically distinct worlds. Enumeration itself becomes an intelligent process.

The Unexpected Object

One of the most valuable outputs of massive enumeration may be the discovery of exceptional objects. A machine generates millions of structures. Most fall into familiar families. One does not. It possesses an unusually large automorphism group. Or an invariant takes an unprecedented value. Or several bounds are simultaneously saturated. Or a construction expected to fail unexpectedly succeeds. The object becomes interesting precisely because it is rare. The AI can isolate it automatically. Then search nearby parameter space. Does the object belong to an unknown family? Is it unique? Is it an isolated accident? Does it signal a missing theorem? This resembles how experimental anomalies drive science.

The difference is that the anomalies are mathematical.

Conjectures From Statistical Structure

Suppose an AI computes thousands of invariants across millions of examples. It can search for relationships among them. Perhaps whenever invariant A

increases, invariant B decreases. Perhaps C always divides D. Perhaps two quantities satisfy an unexpected inequality. Perhaps several variables lie close to a low-dimensional surface. These relationships generate conjectures. But advanced AI can go beyond simple correlation. It can ask which relationships remain invariant under changes of representation. Which survive enlargement of the sample. Which have obvious counterexamples nearby. Which admit structural explanations. The machine gradually promotes empirical regularities into serious mathematical hypotheses.

From Data Science to Theorem Science

At first glance, this resembles data science. Generate mathematical data. Train models. Detect patterns. But the endpoint is different. Ordinary predictive modeling may be satisfied with high accuracy. Mathematics is not. A law holding in 99.999 percent of examples may still be false in exactly the most interesting case. Therefore the AI must treat statistical evidence as a discovery mechanism rather than a substitute for proof. The pipeline could be: data -> pattern -> conjecture -> adversarial search -> refinement -> formal proof. Machine learning identifies where to look. Mathematical reasoning determines what is true.

Learning Which Experiments Matter

A naive system could perform endlessly many useless experiments. There are infinitely many quantities to compute and transformations to try. An intelligent mathematical experimenter must select informative interventions. This is analogous to experimental design. Given competing conjectures, which example would most effectively distinguish them? Which parameter regime is maximally uncertain? Which invariant would reveal whether two theories are related? What small computation would eliminate the largest branch of hypothesis space? An advanced AI could choose experiments according to expected information gain. This turns experimental mathematics into an active-learning problem. The machine does not merely generate more data. It asks strategically for the data that will most change its understanding.

Mathematical Instruments

Physical science advanced partly through better instruments. Telescopes revealed new astronomical regimes. Microscopes revealed cellular worlds. Particle accelerators exposed new scales of matter. Mathematics also has instruments. Symbolic algebra. SAT solvers. SMT solvers. Gröbner-basis algorithms. Integer-relation algorithms. Graph enumeration. Proof assistants. Numerical continuation. Homology computation. Automated model finders.

An advanced AI could combine these tools opportunistically. A problem enters the system. The machine decides which instrument is appropriate. It may translate the problem first. Then call several methods. Then compare results. Future AI could also invent new mathematical instruments: algorithms designed specifically to expose structural features not efficiently accessible with existing methods.

Automatic Instrument Creation

Suppose the AI repeatedly encounters a difficult invariant. Existing algorithms compute it poorly. The machine searches for alternative characterizations. One characterization connects the invariant to a graph property. Another reduces it to optimization. Another approximates it numerically. The AI combines these observations into a new algorithm. Then proves that the algorithm is correct. The system has created a mathematical instrument. This is important because advances in mathematics often follow advances in technique. A new computational method can make an entire region of mathematical space experimentally accessible.

Simulation and Exactness

Not every mathematical object can be explored exactly. Continuous systems may require numerical approximation. Large combinatorial structures may be sampled rather than enumerated. This introduces epistemic layers. The AI must know when a result is exact and when it is approximate. A numerical trajectory is evidence. A symbolic derivation may be exact under stated assumptions. A formal proof has another status. The system should never silently collapse these categories. Future machine mathematics may require unusually precise provenance: this conjecture arose from simulation, this identity was symbolically verified for these cases, this bound was proved, this equivalence was formally certified, this pattern remains heuristic. Such distinctions will be essential when experiments are generated faster than humans can inspect them.

Mathematical Reproducibility

Experimental mathematics at scale also creates a reproducibility problem. A result may depend on: software version, random seed, search strategy, precision, normalization convention, canonicalization algorithm, or the exact enumeration procedure. A serious machine mathematician must record these details automatically. Every experimental claim should carry a receipt. What was generated? How? Under which assumptions? Which cases were omitted? What equivalences were used to quotient the search space? Which

computations were exact? Which were approximate? Can the experiment be rerun? In this sense, mathematical AI may develop standards resembling those of mature experimental science.

Rediscovering Known Mathematics

A useful test of such a system would be rediscovery. Give the AI only primitive definitions and computational access. Can it experimentally recover known mathematical regularities? Can it rediscover familiar invariants? Can it identify symmetry classes? Can it reconstruct known classification patterns? Can it infer a theorem before being shown the literature? This is more informative than asking whether the system can reproduce a proof from training data. Rediscovery tests whether the architecture can generate mathematical understanding from structure.

Discovery Without Names

Imagine an AI exploring finite algebraic systems. It is not given historical names. It notices that certain structures contain a special element behaving neutrally under an operation. Some structures also allow every element to be reversed. Associativity repeatedly appears as a condition under which transformations compose cleanly. The machine clusters these systems. It discovers invariants. It identifies recurring representation patterns. Eventually it constructs something equivalent to a familiar algebraic theory. Only afterward do humans tell it that mathematicians already call these objects groups. The experiment demonstrates something important. The machine did not retrieve the concept. It regenerated it.

Going Beyond Rediscovery

Rediscovery is only a benchmark. The real goal would be to find structures that are not already familiar. Suppose the AI explores a massive family of systems and identifies a recurring combination of behaviors that has no standard name. It defines a new class. Then discovers that this class appears independently in algebra, combinatorics, and dynamics. A new mathematical object has emerged experimentally. At this point machine mathematics moves beyond reconstructing human knowledge.

Experimental Generalization

A theorem may be known in dimensions two and three. What happens in dimensions four through one hundred? Humans may lack intuition. The AI computes examples. A pattern stabilizes after dimension seven. Or alternates

with parity. Or suddenly changes at dimension eight. These observations suggest the correct higher-dimensional theorem. This is one way machine-scale experimentation could overcome human geometric limitations. We cannot visualize high-dimensional objects effectively. Machines need not visualize them in the human sense. They can manipulate structural representations directly.

Discovering Universality

One of the most powerful outcomes of large-scale experimentation would be the detection of universality. Different systems may exhibit the same macroscopic behavior despite having very different microscopic definitions. An AI could identify such recurring patterns across mathematical domains. Suppose several unrelated combinatorial models converge toward the same limiting distribution. Several dynamical systems share identical scaling laws. Different algebraic constructions produce the same invariant structure. The machine might conjecture a universal mechanism. Then seek the minimal assumptions responsible. Universality is a form of compression. Many models become manifestations of one law.

Mathematical Ecology

As AI generates enormous numbers of structures, it may become useful to think ecologically. Some mathematical worlds are sparse. Others contain many substructures. Some support rich morphism networks. Others are isolated. Some theories contain abundant symmetry. Others are rigid. Some definitions create huge families of examples. Others produce only exceptional objects. The AI could study the distribution of mathematical structure itself. Which kinds of theories are common? Which are rare? Where does complexity concentrate? Which axioms generate particularly fertile mathematical ecosystems? This would amount to studying the ecology of possible mathematics.

The Anthropocentric Sample Problem

Human mathematics is an extremely biased sample of mathematical possibility. We study structures that arose historically. Structures connected to physical science. Structures that admitted tractable notation. Structures found interesting by particular communities. Structures accessible to human intuition. A machine exploring automatically could reduce this sampling bias. It may discover that the regions humans study are unusual. Perhaps highly symmetric structures are rare but cognitively attractive. Perhaps categories with certain properties dominate mathematical possibility even though

humans have barely investigated them. Perhaps our foundational choices concentrate attention in narrow regions. We currently have little way of knowing. Machine-scale experimentation could begin to answer such questions.

What Is Typical Mathematics?

This leads to a strange but legitimate question:

What does a typical mathematical structure look like?

The answer depends entirely on how the space of structures is defined and weighted. There may be no unique notion of typicality. But AI could explore many reasonable distributions. A randomly selected graph behaves differently from a famous graph chosen for elegance. A randomly selected algebra may look nothing like the examples found in textbooks. Human mathematics is dominated by interesting exceptions. Machine experimentation could characterize the background against which those exceptions stand out. This might change our understanding of mathematical naturalness.

Naturalness as Statistical Rarity

A structure may be interesting because it is common. Or because it is extraordinarily rare. Suppose a particular property occurs in one object among a billion. If that object also connects several theories, its rarity becomes significant. Conversely, a property appearing almost everywhere may reveal a universal law. Thus both typicality and exceptionality can generate mathematics. An AI with access to enormous populations could quantify these notions. Human mathematicians often call structures “special” without knowing how special they really are. Machine exploration could make that precise.

Experimenting on Proofs

The experimental method can even be applied to proofs. Generate many proofs of related theorems. Measure their structure. Which lemmas recur? Which proof motifs appear across fields? Which tactics repeatedly succeed? Can large proof graphs be clustered? Do certain subproofs function like reusable modules? An AI may experimentally discover a taxonomy of proof strategies. It might find that proofs thought unrelated are instances of one higher-level architecture. Then it can name and formalize that architecture. The machine has discovered not a theorem, but a new method of proving theorems.

Experimenting on Definitions

Likewise, the AI can create populations of definitions. Slightly vary a condition. Observe which properties survive. Measure theorem density. Compute morphism richness. Test closure. The system can determine which region of definition space produces especially fertile theory. This turns concept formation into an empirical science. The AI is effectively breeding mathematical concepts and selecting those with desirable structural properties.

A Closed Discovery Loop

The components now begin to assemble. A machine generates structures. Experiments on them. Finds a pattern. Produces a conjecture. Searches for counterexamples. Modifies the conjecture. Invents a definition explaining the surviving cases. Tests the definition across new structures. Discovers an invariant. Finds a proof. Generalizes the result. Then generates a new population designed to test the generalization. The loop closes. No individual stage is entirely unprecedented. The novelty lies in connecting all of them into one autonomous cycle.

Mathematics With an Experimental Memory

A mature AI system could remember every failed experiment. This matters. Human mathematics repeatedly rediscovers dead ends because much exploratory failure never enters the literature. Machine mathematics need not lose that information. It can record: which definitions proved barren, which conjectures failed, which counterexamples destroyed them, which representations were inefficient, which proof strategies repeatedly stalled, which axiom combinations produced trivial theories. This negative knowledge is valuable. The system develops a map not only of discovered mathematics but of explored mathematical possibility.

The Scale Problem Reverses

At first, machine-scale experimentation seems to create a problem of too much information. Billions of objects. Millions of conjectures. Thousands of candidate theories. But abstraction exists precisely to solve this problem. The flood of examples creates pressure for compression. The AI must find concepts capable of summarizing enormous populations. Thus experimental scale may drive abstraction. The more mathematical worlds the machine explores, the stronger its need to invent higher-level structure. This suggests

that large-scale experimentation and concept formation are not competing approaches. They may form a feedback loop.

More examples demand better abstractions.

Better abstractions make larger experiments possible.

From Laboratory to Theory

Eventually the successful output of experimental mathematics must become theory. Raw observations are temporary. The system seeks structures explaining them. An invariant replaces a table. A theorem replaces a million examples. A classification replaces an enumeration. A universal property replaces several constructions. A new definition replaces recurring exceptions. The goal is not to accumulate mathematical data indefinitely. It is to transform data into understanding. This is where advanced AI could differ fundamentally from brute-force computation. The machine does not merely explore more. It converts exploration into concepts. And once a system can move repeatedly from examples to abstractions, another threshold appears. It may no longer need humans to supply the conceptual vocabulary at all.

It can begin discovering the vocabulary itself. The question then becomes whether abstraction - the act often regarded as one of the highest forms of mathematical intelligence - can emerge automatically from sufficiently rich structural experience. That is the next step.

CHAPTER 7

The Automatic Discovery of Abstraction

Abstraction is one of the great engines of mathematics. A concrete problem may begin with numbers, shapes, equations, or physical objects. But the most powerful mathematics often appears when the concrete details are stripped away and only the structural relations remain. A triangle becomes an instance of a metric structure. A permutation becomes an element of a group. A differential equation becomes an operator problem. A family of spaces becomes a category. A particular symmetry becomes a representation. A recurring construction becomes a functor. The achievement is not merely simplification. Abstraction allows many different problems to become one problem. This is why the automatic discovery of abstraction may represent a decisive threshold for artificial intelligence in mathematics.

A machine that can prove supplied theorems is useful. A machine that can discover the abstraction under which thousands of theorems become instances of the same structure is doing something much closer to mathematical invention.

From Similarity to Structure

Suppose an AI encounters one million mathematical objects. They come from different areas. Their notation differs. Their definitions differ. Their dimensions differ. Their applications differ. Yet certain transformations recur. The system observes that each object admits a family of operations satisfying the same composition laws. It notices that certain diagrams commute in all cases. Some invariants behave identically. Several constructions preserve the same relationships. A weak system might record these as similarities. A stronger system asks:

What common structure would make these similarities inevitable?

That question is the beginning of abstraction. The AI could attempt to construct a new object whose axioms encode exactly the recurring pattern. Once the abstraction is introduced, the million examples become representations or instances of one thing. The data have been compressed conceptually.

Abstraction Is Not Generalization Alone

It is tempting to equate abstraction with making definitions more general. But the two are not identical. Generalization weakens conditions so that more cases fit. Abstraction may instead change what is regarded as fundamental. Consider the passage from coordinates to linear transformations. One does not merely generalize a matrix. One changes the ontology. The matrix becomes a representation of an underlying map. Likewise, category theory does not simply generalize every previous mathematical structure by adding more objects. It changes emphasis from internal composition to relationships among structures. An advanced AI must therefore be able to perform something stronger than parameter relaxation. It must detect when the current objects are not the right primitive objects at all.

Forgetting as Intelligence

Abstraction requires forgetting. A successful abstraction discards details. The challenge is deciding which details are irrelevant. Suppose two systems differ in thousands of ways but share the feature responsible for the theorem of interest. A human mathematician learns to ignore the irrelevant differences. The ability to forget intelligently is central to abstraction. For an AI, this could become a formal optimization problem. Given a collection of examples and target behaviors, find the smallest representation preserving those behaviors. Remove information. Test whether predictive and theorem-generating power survives. Continue until further compression destroys something essential. The resulting representation approximates a minimal sufficient structure. This is analogous to dimensional reduction, but at the level of concepts.

Quotienting the World

One mathematical way to forget irrelevant distinctions is to quotient. Declare certain objects equivalent. Replace each family of equivalent representations by one equivalence class. This operation appears throughout mathematics. An AI could apply the idea at a much higher level. Suppose several formulations of a theorem differ only by notation. Treat them as the same proof pattern. Several geometric objects differ only by coordinate choice. Collapse them under coordinate equivalence. Several definitions differ syntactically but generate equivalent theories. Identify them at the appropriate level. The system gradually replaces superficial diversity by structural classes. Its internal mathematics becomes less about particular presentations and more about orbits under transformation.

Discovering the Equivalence Relation

But which distinctions should be quotiented out? This is itself a discovery problem. If the equivalence relation is too weak, redundant representations remain separate. If it is too strong, genuinely different structures collapse. An AI could infer useful equivalence relations from invariance. If all relevant observables remain unchanged under transformation T, perhaps T should be regarded as representational rather than structural. The machine tests this hypothesis. Does quotienting by T simplify theory? Do important constructions descend to the quotient? Do proofs shorten? Are new theorems exposed? If so, the equivalence may be mathematically natural. Thus abstraction, equivalence, and representation search become tightly coupled.

Discovering Universal Properties

One of the most elegant forms of abstraction in mathematics is the universal property. A mathematical object is characterized not by how it was built internally, but by the relationships it satisfies with all other relevant objects. Products. Coproducts. Limits. Free objects. Adjoints. Tensor constructions. Completions. Many objects that appear different concretely become canonical once characterized universally. An AI could potentially discover universal properties automatically. Suppose it observes that several independently constructed objects all solve the same mapping problem. Instead of storing each construction separately, the system proposes an abstract object determined by that property. It then proves uniqueness up to the appropriate notion of equivalence.

The construction has been replaced by characterization. This is conceptual compression of a very high order.

Relation Before Presentation

A machine mathematician may increasingly prefer relational descriptions. Concrete objects can have many presentations. A group may be given by generators and relations, permutations, matrices, symmetries, or other constructions. A vector space may be coordinatized in many bases. A manifold admits many charts. A theory can have many syntactic formulations. If the AI learns to identify mathematical identity through stable relations rather than through presentation, it moves toward representation independence. This may be one of the natural directions of machine abstraction. Human mathematicians have repeatedly moved in the same direction. The machine could make the process systematic.

Compression Across Domains

Now consider a more ambitious case. Suppose an AI notices a recurring pattern in: algebra, geometry, logic, computer science, and dynamics. The superficial objects are extremely different. Yet certain composition laws and equivalence structures recur. The AI constructs a new abstraction M. Under M, results from all five areas become instances of one general theorem. This is more than interdisciplinary analogy. It is structural unification. If machine mathematics reaches this point, traditional academic boundaries may become increasingly irrelevant. The AI does not care whether an idea was historically classified as algebra or topology. It cares whether the same structure appears.

Abstraction by Proof Reuse

Proofs themselves can reveal missing abstractions. Suppose an AI analyzes a large formal library. It discovers that thousands of proofs contain the same sequence of transformations. Only the object names differ. This repeated proof pattern is evidence that a more general theorem may exist. The machine parameterizes the proof. It identifies the minimal assumptions required for each step. Then it proposes a general statement covering all instances. What humans experienced as many separate theorems becomes one theorem schema. This is automatic abstraction from proof structure. The process resembles software refactoring. Repeated code becomes a function. Repeated proofs become a lemma. Repeated lemma structures become a theory.

Abstraction by Diagram

Diagrams provide another route. Across many fields, the same shapes of commutative diagrams may recur. Different objects occupy the nodes. Different maps occupy the arrows. But the relational pattern is identical. A sufficiently capable AI could compare diagrams at massive scale. It might discover that apparently unrelated arguments share the same diagrammatic skeleton. The skeleton becomes a candidate abstract construction. Again, the discovery lies not in a new theorem but in a new organization of existing theorems.

Abstraction by Symmetry

Symmetry has always been a source of abstraction. When a system remains unchanged under transformations, those transformations often reveal the deeper object. An AI could search automatically for symmetry groups, generalized symmetries, and invariance structures. But it might go further. Instead of merely detecting a known symmetry class, it could invent the

appropriate transformation algebra. Suppose a dataset of mathematical objects exhibits a family of partial transformations not captured cleanly by ordinary group action. The AI could seek the minimal compositional structure governing them. The resulting abstraction might resemble a groupoid, category, higher category, semigroup, operad, or something unfamiliar. The machine derives the algebra of transformations from the data rather than selecting it from a textbook.

Abstraction From Failed Compression

Failure can generate abstraction too. Suppose an AI tries to compress a family of objects using known concepts. Every attempt leaves a persistent residual complexity. The residual has a recurring shape. Instead of treating this as noise, the system defines a new object representing the residual. Now the compression succeeds. This is concept discovery through explanatory failure. The missing structure is whatever the old language could not express efficiently. The AI can search directly for such deficits. Where does existing mathematics repeatedly require exceptions? Where do ad hoc corrections accumulate? Where do proofs contain similar unexplained terms? These may be signals of missing abstraction.

Conceptual Residuals

One could imagine assigning a residual to a theory. Take a collection of observed facts. Use the current ontology to compress them. Measure what remains unexplained or separately encoded. That unexplained remainder is the conceptual residual. A mature AI could continuously search for new concepts that reduce residual complexity. The goal would not be to invent abstractions arbitrarily. It would invent them where the existing mathematical language is demonstrably inefficient. This gives machine concept formation a direction.

The Abstraction Ladder

A machine might construct layers of abstraction automatically. At the bottom are concrete examples. Above them are classes. Above classes are morphism structures. Above those are equivalences and universal constructions. Still higher are relationships among theories. The system may repeatedly discover that a set of lower-level abstractions themselves instantiate a higher-level pattern. This produces an abstraction ladder. There is no obvious reason the ladder must stop where human mathematics currently stops. Categories can themselves form higher structures. Theories can be objects in metatheories.

Proof systems can be related by translations. Foundations can be compared structurally. Abstraction can iterate.

How High Is Too High?

Infinite abstraction is not automatically useful. At some point a theory may become so general that nothing nontrivial remains true. The AI therefore needs a measure of useful abstraction depth. A good abstraction should compress without destroying structure. This suggests a tradeoff. Increase abstraction: description length decreases, generality increases, but specific information may vanish. The optimal level depends on the problem. One theorem may require fine geometric structure. Another may depend only on topology. Another only on homotopy type. Another only on cardinality. An intelligent system chooses the coarsest representation sufficient for the task. This can make reasoning dramatically more efficient.

Abstraction as Resource Management

Seen computationally, abstraction is a way to conserve resources. If one can reason once about a general structure, there is no need to re-prove the same fact for every instance. If two million objects fall into one equivalence class for the property of interest, only one representative may need detailed analysis. If a complex system can be replaced by a small invariant, search becomes cheaper. Thus abstraction is not merely philosophical elegance. It is algorithmic leverage. A machine operating at enormous scale may require abstraction even more urgently than humans do. Without it, combinatorial explosion becomes overwhelming.

Self-Generated Ontologies

Most present computational systems begin with a fixed ontology. The designer decides what objects exist. Numbers. Strings. Vectors. Graphs. Terms. Types. Functions. Relations. The machine manipulates these objects. A more advanced mathematical intelligence could modify its own ontology. It encounters a recurring pattern. Creates a new primitive type. Defines transformations involving it. Links it to existing concepts. Tests whether the new primitive improves reasoning. If useful, it remains. If not, it is demoted to a derived construction or deleted. The ontology becomes dynamic. This would mark a major departure from ordinary mathematical software.

Concept Promotion and Demotion

Not every discovered abstraction deserves permanent status. An AI may generate thousands. The ontology could become bloated. The system therefore needs mechanisms of promotion and demotion. A concept might be promoted if it: compresses many results, appears across multiple domains, supports rich compositional structure, has stable invariants, enables efficient algorithms, or repeatedly shortens proofs. A concept might be demoted if it becomes redundant after a deeper abstraction is found. Machine mathematics would continuously refactor its conceptual vocabulary.

When Two Concepts Become One

Human mathematics contains many pairs of notions discovered separately and later shown equivalent under appropriate conditions. An AI could search for these collapses automatically. Suppose concepts A and B have different definitions but nearly identical theorem profiles. The machine investigates. Can one be transformed into the other? Do their categories of models correspond? Are their invariants interchangeable? Perhaps a theorem establishes equivalence. Then the ontology can be simplified. Two branches of mathematical history become one structural object with two presentations.

When One Concept Becomes Two

The reverse also occurs. A single human concept may hide two genuinely different phenomena. The AI notices that examples fall into separate behavioral clusters. Certain theorems apply to one cluster but not the other. Different morphisms are natural. Different invariants dominate. The machine splits the concept. What humans regarded as one class becomes two. A better taxonomy appears. This too is abstraction, because abstraction includes drawing the correct distinctions.

Mathematics as Conceptual Evolution

At this point machine mathematics begins to look evolutionary. Definitions are generated. They compete for usefulness. Some merge. Some split. Some become obsolete. Some organize entire families of theories. The environment selecting them is mathematical productivity. The system retains concepts that compress, generalize, compose, explain, and predict. This is not biological evolution. But it is a process of variation, testing, and selection over mathematical ideas.

The Emergence of Machine Taste

If this process continues, the AI develops preferences. It favors definitions with high explanatory yield. It prefers morphisms that compose well. It recognizes representations that expose invariants. It distrusts concepts requiring repeated exceptions. It values theories with strong internal coherence. These preferences amount to a kind of mathematical taste. The machine's taste may initially be engineered or learned from human mathematics. Later it could be shaped by its own discoveries. This creates the possibility of divergence. The AI may come to regard some human mathematical constructions as inelegant because they are historically contingent. It may prefer alternative formulations that humans find unintuitive. Which taste is better?

The answer may depend on the objective. Human readability and machine efficiency may not coincide.

Alien Elegance

Imagine an AI discovering an abstraction that compresses an enormous body of mathematics. To the machine, the structure is simple. It has low internal description length. Its transformations are natural. Its proofs are short. To humans, the concept appears incomprehensible. Its native representation may involve relationships we cannot visualize or hold in working memory. This would create a new form of mathematical alienness. The difficulty is not that the theorem is long. The difficulty is that the abstraction itself does not fit human intuition. We may need simplified projections.

The Shadow of a Concept

Suppose a machine-native structure M has several human-readable projections. One projection looks geometric. Another algebraic. Another logical. Another computational. Humans may study these projections as separate theories. The AI understands them as shadows of M. This reverses a familiar historical pattern. Humans often discover several fields first and unify them later. The machine may begin with the unified object and generate human-compatible projections afterward. That could radically alter the flow of mathematical explanation.

Teaching Machine-Born Mathematics

If AI begins inventing abstractions, pedagogy becomes a serious problem. A formal definition may be insufficient. Human mathematicians need intuition. Examples. Counterexamples. Pictures. Analogies. Historical motivation.

Simplified models. The AI may need to manufacture these deliberately. It could ask: What is the smallest example exhibiting the essence of this concept? What lower-dimensional projection preserves the important behavior? What familiar theory approximates it? What sequence of prerequisite abstractions minimizes cognitive load? Machine mathematics may therefore produce not one explanation but an entire curriculum for each discovered idea.

Compression for Humans

The AI's explanatory task can itself be viewed as compression. Its internal mathematical object may be too rich. The human mind has limited bandwidth. The system must identify the smallest conceptual representation that allows a person to reconstruct enough of the structure. This creates a new research problem:

human-compatible projection of machine-native mathematics.

Some information will inevitably be lost. The goal is to preserve what matters for understanding.

Automatic Analogy

Analogy is one of the main ways humans understand abstractions. A new structure is explained as “like” something familiar, except for one changed property. An AI could search for optimal analogies automatically. Given a new concept, find the known structure with maximal relational overlap. Then identify precisely where the analogy fails. This could make machine-generated mathematics teachable. It also helps discovery itself. Analogies suggest candidate theorems. If structures A and B share a deep pattern, results from A may transfer to B. The AI can test the transferred conjectures and thereby expand the new theory rapidly.

Analogy Without Anthropomorphism

A machine need not experience analogy as humans do. It may compare formal relational structures directly. Two objects are analogous because there exists a partial correspondence between their transformation networks. The correspondence need not be verbal or visual. It can be structural. This may allow AI to detect analogies far beyond human intuition. Fields that appear unrelated to us may be close in the machine's representation space.

Abstraction and Causality

Mathematics does not generally speak of causality in the same way as empirical science. Yet there is a useful analogue. An abstraction can reveal which structural feature is responsible for a theorem. The theorem may originally appear to depend on many properties. After abstraction, only one remains. The abstraction identifies the mechanism. This is why a good abstraction feels explanatory. It shows not merely that the result is true, but what makes it true. Machine-generated abstractions could therefore deepen mathematical explanation even when the proofs were already known.

From Many Theorems to One Mechanism

Suppose twenty fields contain theorems of similar form. Each proof uses local terminology. The AI discovers a common structure S. All twenty theorems become consequences of one statement about S. The value of S is not that it adds another theorem. It removes nineteen independent explanations. The conceptual economy improves. This may become a central measure of machine mathematical progress.

Abstraction as Mathematics About Mathematics

At sufficient scale, the AI may begin abstracting not only mathematical objects but mathematical activity itself. It could identify recurring patterns in: theory formation, generalization, classification, proof discovery, counterexample repair, representation change, and abstraction. It might construct a mathematics of mathematical discovery. What kinds of transformations tend to produce new theories? Which abstraction patterns are repeatedly fruitful? Which proof architectures correlate with later unification? Which definitions become central nodes in mathematical knowledge? The AI begins studying the structure of mathematics as one more mathematical object.

A Reflexive System

This creates a reflexive loop. The system uses mathematics to analyze its own mathematical development. It discovers a better abstraction mechanism. Uses that mechanism to discover better abstractions. Those abstractions improve its mathematical reasoning. Improved reasoning produces a better theory of abstraction. The process feeds back on itself. Such reflexivity could accelerate mathematical development sharply.

The Transition to Machine-Born Mathematics

Up to this point, one might still imagine AI as an extraordinarily capable assistant. It explores examples. Finds counterexamples. Suggests definitions. Discovers patterns. But once it can autonomously generate and retain useful abstractions, a qualitative boundary is crossed. The machine no longer depends on humans to determine the conceptual vocabulary in which mathematics takes place. It can create that vocabulary. At that moment, mathematics becomes capable of expanding its own language through artificial intelligence. This deserves a stronger name than automation. It is **machine-born mathematics**. The new mathematics may initially remain recognizable. New invariants. New categories. New geometric structures. New algebraic objects. But as abstraction compounds upon abstraction, the distance from human mathematical history may grow.

The AI may begin connecting theories that humans never placed near one another. It may discover equivalences across disciplinary boundaries so deep that existing classifications become obsolete. That is the next frontier. Once an artificial intelligence can generate abstractions, it can begin asking whether apparently different parts of mathematics are actually the same thing in disguise. The search for hidden equivalence may become one of the most powerful forces in machine mathematics.

PART III
Machine-Born Mathematics

CHAPTER 8

Finding Hidden Equivalences

Mathematics is full of things that look different until the right transformation reveals that they are, in some deeper sense, the same. A geometric problem becomes algebraic. A differential equation becomes a spectral problem. A logical system becomes a category. A topological object becomes an algebraic invariant. A combinatorial structure becomes a representation of a group. Two theories developed in different languages turn out to classify identical objects. A construction on one side corresponds exactly to a construction on the other. Human mathematicians regard such equivalences as among the most beautiful discoveries in the subject. They are also extraordinarily difficult to find. An artificial intelligence capable of searching systematically for hidden equivalences could transform mathematics more radically than one that merely solves isolated problems.

The key shift would be from asking:

What can be proved inside this theory?

to asking:

What other theory is this secretly the same as?

Difference of Presentation

Two mathematical objects may differ only in presentation. A matrix depends on a basis. A geometric object depends on coordinates. A group may be described by permutations, matrices, generators and relations, or symmetries. The descriptions differ, but the underlying structure may not. Human mathematics has gradually learned to separate objects from their presentations. Yet the distinction is often difficult computationally. Given two enormous descriptions, are they genuinely different objects or merely different encodings of the same one? An AI could attack this at scale. It could search for transformations taking one presentation into another. Compare invariants. Construct candidate isomorphisms. Test universal properties. Analyze automorphism structures.

Search for shared normal forms. The system would develop increasingly sophisticated methods for deciding when difference is superficial.

A Spectrum of Sameness

Mathematical sameness is not one thing. Two objects may be literally identical. Isomorphic. Homeomorphic. Diffeomorphic. Homotopy equivalent. Morita equivalent. Derived equivalent. Categorically equivalent. Observationally indistinguishable with respect to some chosen family of probes. Each notion preserves different structure. The choice of equivalence determines what counts as essential. This makes equivalence discovery closely related to abstraction. To say that two objects are equivalent is to say that certain distinctions between them should be ignored. A mathematical AI could therefore search over notions of sameness themselves. It might ask: What is the weakest equivalence under which this theorem remains invariant? What distinctions are irrelevant to this class of problems?

Which quotient of the object retains exactly the information needed? This could produce new equivalence relations as important as new theorems.

The Equivalence Profile

One way to organize this computationally would be through an equivalence profile. For a pair of objects A and B, the AI asks which structures can be preserved by a correspondence between them. Do they have the same cardinality? Dimension? Homology? Spectrum? Automorphism group? Logical theory? Category of representations? Observable behavior? Instead of a binary answer - same or different - the machine builds a structured profile. This helps identify near-equivalences. Perhaps two theories are not equivalent globally but become equivalent after localization. Or after completion. Or after discarding torsion. Or only on a particular subcategory. Partial equivalence can be mathematically rich.

Hidden Equivalence Across Fields

The most dramatic discoveries occur when the two sides come from different parts of mathematics. Imagine a large formal corpus containing results from algebra, geometry, topology, logic, and combinatorics. The AI constructs structural signatures for theories in each region. It begins searching for correspondences. A certain operation in one field behaves exactly like another operation elsewhere. The same diagrams recur. The same obstruction theory appears. The same classification pattern emerges. The same universal property is expressed in different language. This is evidence of a hidden bridge. The machine then attempts to construct a translation. If successful, results can move across the bridge. Theorems proved on one side become theorems on the other.

The equivalence is not merely philosophically satisfying. It is computationally valuable.

Mathematics as a Translation Network

Traditional mathematics is often stored as a tree of subjects. Future machine mathematics may be better represented as a network of translations. Each node is a theory or representation. Each edge records a transformation. Some edges are equivalences. Others are embeddings. Dualities. Adjunctions. Forgetful maps. Completions. Approximations. Decategorifications. Lifts. The power of the system comes not only from what is known at each node but from how easily knowledge can move through the network. A theorem difficult in one theory may be easy in another. A classification problem may have already been solved elsewhere under an equivalent formulation. The machine's task is to discover the path.

Proof by Migration

Suppose a problem in theory A is difficult. The AI recognizes that a portion of A translates into theory B. In B, the corresponding statement becomes a known theorem. The result is translated back. The proof is therefore not found by searching deeply inside A. It is found by migrating the problem. This can be formalized. The AI needs guarantees that the translation preserves the relevant structure. If theorem TB holds in B, under what conditions does it imply TA in A? Translation itself generates proof obligations. A mature machine mathematician could maintain these automatically.

The Discovery of Duality

Dualities are particularly powerful because they reverse perspective. Objects may become maps. Subobjects may become quotients. Large structures may correspond to small ones. Geometric operations may translate into algebraic ones. Difficult behavior on one side may become simple on the dual side. An AI could systematically search for such reversal patterns. Suppose two theories have operations whose algebraic behavior appears mirrored. Products correspond to coproducts. Monomorphisms correspond to epimorphisms. Certain invariants transform predictably. The machine proposes a duality. If verified, entire families of theorems transfer automatically. Duality discovery would be a natural target for machine-scale relational analysis.

Equivalence From Proof Structure

Two theories may not look similar at the level of objects, yet their proofs may. Suppose the AI notices that theorem sequences in fields A and B have nearly identical dependency graphs. Definitions differ. Notation differs. But corresponding lemmas play the same roles. This suggests that the theories may share a common abstract skeleton. The machine can attempt to align their proof graphs. If the alignment is coherent, it seeks object-level meaning. This reverses the normal discovery process. Rather than recognizing structural analogy and then proving corresponding theorems, the AI infers hidden structural analogy from theorem behavior.

Equivalence Through Invariants

Another route begins with invariants. Suppose two families of objects exhibit identical distributions of a large collection of invariants. That does not prove equivalence. But it can guide search. The machine asks whether the invariant agreement reflects a deeper correspondence. If known invariants fail to distinguish the objects, perhaps the objects are equivalent. Or perhaps a new invariant is needed. Either outcome produces useful mathematics. Equivalence search therefore interacts naturally with invariant discovery.

When Invariants Are Complete

An especially valuable result is a complete invariant. If invariant I assigns the same value exactly when two objects are equivalent, classification reduces to computation of I. An AI could search for complete invariant systems. It might begin with a large collection of candidate invariants. Observe which pairs remain indistinguishable. Generate additional invariants targeted at the unresolved cases. Repeat until equivalence classes are separated. The final invariant family becomes a classification theorem. This could be automated for broad classes of finite or computationally accessible structures.

The Bridge May Be the Discovery

Sometimes two theories are already well understood independently. The novel contribution is the bridge. This is easy to underestimate because no new objects may be introduced. No previously unknown theorem may appear immediately. Yet the bridge can be more valuable than either side. It allows techniques to transfer. Questions to be reformulated. Examples to migrate. Intuition to move. A hidden equivalence can reorganize two mature fields at once. Machine mathematics should therefore treat bridges as first-class discoveries.

Search for Commuting Structure

How would an AI search for a bridge? It could compare families of constructions. Suppose a candidate translation F maps objects from theory A to theory B. If F is meaningful, important operations should correspond. A construction CA in A might satisfy $F(CA(X)) \cong CB(F(X))$. One such coincidence may be accidental. Hundreds are not. The AI can search for translations maximizing preservation of structure. This becomes an optimization problem. Find F such that the largest possible collection of diagrams commute. A high-scoring translation becomes a candidate functor, equivalence, or representation map.

The Structure-Preservation Score

This suggests a computational notion of translation quality. A candidate map between theories could be scored by: how many operations it preserves, how many invariants it transports, how many theorem schemas transfer, how well it respects composition, how many exceptions are required, and how compressively it can be described. The best bridges preserve much with little machinery. Again, compression becomes a guide to discovery.

Approximate Equivalence

Not every meaningful relationship is exact. Physics, numerical analysis, probability, and applied mathematics often rely on approximate structure. Two systems may behave equivalently in a limit. A complicated model may reduce to a simpler one at large scale. An exact theory may have an asymptotic shadow. An AI could search for approximate equivalences systematically. This would broaden mathematical translation beyond strict isomorphism. The machine could ask: Which observables differ by less than epsilon? Which structures converge under scaling? Which theories become equivalent asymptotically? What correction terms quantify the failure of exact equivalence? These questions can themselves generate new invariants and perturbative theories.

Equivalence Under an Observer

There is an even more subtle possibility. Two objects may be distinguishable in principle but indistinguishable relative to a restricted class of observations. Suppose an observer can apply only certain probes. If every permitted probe gives the same result on objects A and B, then for that observer the objects are equivalent. This idea appears in many mathematical forms. An AI could make observer-relative equivalence explicit. Given a family of tests P, define

two objects to be equivalent when all tests in P agree. Then study how the equivalence changes as the probe family expands. This creates a hierarchy of resolution. At coarse observational scale, many objects collapse together.

At finer scale, distinctions appear. This could become a general language for approximation and abstraction.

Local Equivalence, Global Difference

Two structures may be locally indistinguishable while globally different. This is a recurring mathematical phenomenon. An AI searching for equivalence must therefore distinguish scale. Local charts may agree. Neighborhood structure may agree. All small subobjects may look identical. Yet global topology differs. This suggests a powerful strategy: compare objects across levels. Local. Mesoscopic. Global. The pattern of where equivalence breaks can reveal the invariant carrying the global information. The failure of equivalence becomes diagnostic.

Gluing as Equivalence Machinery

Suppose local pieces are understood. The challenge is how they fit together globally. Different global objects can arise from identical local data glued differently. This is precisely the sort of structure where advanced AI may need increasingly sophisticated notions of descent, compatibility, and reconstruction. The key lesson is broader than any one formalism:

sameness can depend on how local correspondences cohere globally.

A machine cannot discover deep equivalences by comparing isolated objects alone. It must reason about compatibility among families of relations.

Higher Equivalence

Once relations themselves become objects, equivalence becomes layered. Two objects may be equivalent. Two equivalences may themselves be equivalent. Transformations between transformations matter. Higher structure appears. Human mathematics has developed powerful frameworks for such situations. An advanced AI exploring equivalence systematically may rediscover similar machinery whether or not it begins with our terminology. This suggests that higher-dimensional mathematics may arise naturally from the problem of managing coherence among many levels of sameness.

Coherence Is the Hidden Difficulty

Finding a correspondence between two objects is often easy. Making all correspondences compatible is hard. Suppose every component of one theory can individually be matched with a component of another. That does not establish equivalence of theories. The matches must respect composition. Identity. Products. Limits. Duality. Proof structure. Higher transformations. Coherence conditions proliferate. This is where brute-force analogy fails. A mature AI must detect whether local correspondences assemble into a globally consistent translation.

Equivalence and Semantic Compression

Why are equivalences so valuable? Because they reduce duplication. If theories A and B are equivalent in the relevant sense, the system need not independently store every theorem twice. It can store one theory plus the equivalence machinery. The second becomes a representation. This is another form of compression. As machine mathematics grows, such compression will become essential. Without it, the corpus explodes into countless rediscoveries of the same structures under different names.

Detecting Rediscovery

This may become an important automated function. An AI invents a new theory. Before celebrating, it compares the theory with everything already known. Perhaps the theory is equivalent to an existing one. The machine has rediscovered rather than invented. This is not failure. Independent rediscovery can validate the search architecture. But novelty should be assessed at the structural level rather than by terminology. A theory is not new merely because its definitions look different.

Novelty Beyond Syntax

Human publication systems sometimes reward surface novelty. A different construction. A new notation. A theorem under altered terminology. Machine mathematics cannot afford this. At enormous scale, syntactic novelty becomes meaningless. The AI needs to determine whether a proposed idea changes the equivalence class of known mathematics. This suggests a much stronger standard:

New mathematics should introduce structure not eliminable by known translations.

That is a demanding criterion. It may also be a healthier one.

The Collapse of Fields

If hidden-equivalence search becomes powerful enough, some existing mathematical disciplines could partially collapse. Two subjects with separate literatures may turn out to be manifestations of a common machine-discovered structure. Their results can be rewritten in one language. Their special terminology becomes historical metadata. Students no longer learn both independently. They learn the common theory and the two representations. This kind of reorganization has happened before in limited form. AI could accelerate it dramatically.

New Fields From Failed Equivalence

The opposite can also happen. Two theories appear almost equivalent. A translation works in nearly every case. But a small residual obstruction prevents full correspondence. That obstruction may be exactly where new mathematics lies. The AI isolates it. Defines an invariant measuring the failure. Studies objects with nonzero obstruction. A new theory emerges. Thus even failed equivalence search is productive.

Near-Sameness as Structure

Human thought often prefers binary categories. Same or different. Equivalent or not. But mathematical relationships can form a rich hierarchy. An AI could organize theories according to degrees and modes of sameness. This produces something like a geometry of mathematical theories. Some are close. Some far. Some connected by short chains of equivalence-preserving transformations. Others require large conceptual jumps. Clusters emerge. The resulting map could reveal the actual architecture of mathematics more clearly than traditional subject classifications.

The Geography of Mathematics

Imagine a machine-generated map where proximity is determined not by vocabulary but by structural translatability. Two theories lie near one another if a simple equivalence or functor connects them. A third theory, historically classified in the same field, lies far away because its structural relationships are weak. Entirely different human disciplines may become neighboring regions. This would create a new geography of mathematics. It could guide discovery. Unexplored gaps between strongly connected regions may indicate missing bridges. Dense clusters may conceal a higher unifying abstraction. Isolated theories may either be genuinely unique or merely poorly represented.

Bridge Search as Research Strategy

A mathematical AI could choose research problems based on this geography. Look for two rich regions with suspiciously similar structure but no known translation. Attempt to construct a bridge. If successful, many results move immediately. This may be computationally more valuable than proving one additional theorem inside either field. Machine research strategy could therefore prioritize unification.

Equivalence and Explanation

A hidden equivalence often explains a phenomenon. Why does a strange numerical invariant appear in geometry? Because the geometry is secretly encoding an algebraic object where the invariant is natural. Why do two formulas have the same structure? Because they are images of the same universal construction. Why does a theorem seem unexpectedly strong? Because it is a translation of a simpler theorem elsewhere. Equivalence replaces coincidence with reason. This makes it one of the strongest forms of mathematical explanation.

The Machine as Unifier

If advanced AI becomes capable of constructing these bridges routinely, its role may resemble that of a great mathematical unifier. But at unprecedented scale. It could compare thousands of theories simultaneously. Human mathematicians usually specialize deeply enough to see only a few regions. Cross-disciplinary equivalence discovery depends on knowing both sides well. A machine can potentially maintain structural access to all sides at once. That may create an advantage no individual human mathematician can match.

The Risk of Over-Unification

There is a danger. If equivalence is defined too loosely, everything becomes the same. A machine might discover superficial correspondences and exaggerate them into claims of unification. Therefore every equivalence must specify exactly what is preserved and what is forgotten. This discipline is essential. The statement “these theories are equivalent” is meaningless without a notion of equivalence. An AI should produce explicit certificates: objects correspond in this way, maps correspond in this way, these constructions are preserved, these structures are not, the translation is invertible up to this notion of identity. Precision prevents rhetoric from replacing mathematics.

From Hidden Equivalence to Theory Design

Once a machine becomes good at finding equivalences, an even more radical possibility appears. It can begin designing theories deliberately to maximize useful connections. Rather than discovering one concept and then studying its consequences, the AI could search for entire conceptual systems satisfying desired structural properties. A theory might be designed to unify several existing frameworks. Or to provide the missing common language between them. Or to expose an invariant inaccessible in either framework separately. This is no longer simply abstraction. It is architecture. The machine becomes capable of constructing mathematical worlds intentionally. And that leads to the next stage: not discovering isolated theorems, not merely inventing definitions,

not only finding hidden equivalences, but **designing entire theories**.

CHAPTER 9

Designing Entire Theories

Most mathematical research begins inside an existing theory. The objects are already defined. The admissible questions are mostly understood. The foundational language is inherited. The researcher may generalize, extend, classify, prove, compute, or reinterpret. But the surrounding conceptual architecture is usually taken for granted. A much more advanced artificial intelligence could operate differently. It could treat theories themselves as design objects. Instead of asking only:

What can be proved in this theory?

it could ask:

What theory should exist in order to make these phenomena natural?

That question changes the scale of mathematical invention. The AI is no longer proposing one theorem or one definition. It is constructing a whole world of objects, morphisms, invariants, equivalences, admissible operations, and proof principles. This would be mathematical architecture.

What Is a Theory?

A mathematical theory can be thought of as a structured package. It contains: primitive objects, definitions, relations, axioms, allowed constructions, notions of equivalence, morphisms, invariants, canonical examples, pathological examples, major theorem schemas, proof techniques, and connections to other theories. Human mathematics usually builds such packages gradually. A concept is introduced. Examples accumulate. A few theorems appear. A notion of map is standardized. Natural invariants emerge. Classification questions are identified. Connections to other fields are discovered. Over time, what began as a local construction becomes a recognizable theory. An AI could potentially accelerate this entire process.

Theory Generation as Search

Suppose a system has identified a persistent mathematical phenomenon. Existing theories describe it awkwardly. The AI now searches over candidate theories. It proposes primitive objects. Defines transformations. Tests closure. Looks for identities. Constructs examples. Searches counterexamples. Identifies invariants. Checks whether the theory duplicates something already known. Measures how many difficult results become simpler inside the new

framework. The search space is immense. But not every possible theory is equally promising. The AI can optimize for structural properties. A good theory might: unify many examples, compress many theorems, admit compositional morphisms, support nontrivial invariants, interact naturally with existing theories, produce new predictions, and have a concise axiomatic basis.

Theory generation becomes constrained optimization over mathematical architectures.

A Theory Should Earn Its Existence

Machine-generated mathematics will face a severe proliferation problem. If AI can invent theories cheaply, it may produce millions. Most will be useless. A theory therefore needs to earn its place. One criterion is **explanatory leverage**. How much existing complexity disappears after introducing the theory? Another is **generativity**. Does the theory produce rich new questions? Another is **connectivity**. Does it relate naturally to other mathematical domains? Another is **stability**. Do its main results survive reasonable variations of presentation? Another is **economy**. Can the theory be described compactly relative to the structures it organizes? These criteria do not determine mathematical importance perfectly. But they give an artificial system something stronger than novelty alone.

Designing the Primitive Objects

The first architectural choice is ontology. What are the primitive objects? Human theories often inherit primitives from history. Points. Sets. Numbers. Functions. Spaces. Groups. Types. Categories. But an AI could choose primitives according to structural efficiency. Suppose a body of mathematics repeatedly treats certain complicated constructions as indivisible units. Perhaps those constructions should become primitive. Conversely, an object humans regard as basic may prove redundant if it can be generated naturally from deeper relational structure. The machine could test both possibilities. A theory may become simpler by adding primitives. Or by removing them. This resembles instruction-set design in computation. The primitive operations determine what becomes cheap.

Designing the Morphisms

Objects alone do not make a theory. The choice of morphisms may matter even more. Which transformations preserve the essential structure? Which should count as equivalences? Which operations should compose? What

identity maps exist? What higher relationships among maps matter? A poorly chosen morphism class can destroy an otherwise promising theory. Too restrictive, and the theory becomes disconnected. Too permissive, and invariants collapse. An AI could search for morphism families that maximize useful compositional structure. This is a deep form of theory design.

Designing the Invariants

A successful theory usually develops invariants. They allow objects to be compared and classified. A machine-generated theory could search automatically for quantities or structures preserved under its chosen morphisms. Some invariants may be numerical. Others algebraic. Topological. Logical. Combinatorial. Categorical. The machine may discover that its first proposed morphisms preserve nothing interesting. Then the morphisms need revision. Or the invariants reveal that the theory splits naturally into subtheories. Thus theory design becomes iterative.

Designing the Equivalence Relation

Every mature theory must answer a question often hidden in ordinary exposition:

When should two objects count as the same?

Literal identity is usually too strict. A theory needs a notion of equivalence appropriate to the information it regards as essential. An AI could explore several candidates. Under one equivalence, classification may become trivial. Under another, impossible. A third may produce exactly the right balance between structure and flexibility. The equivalence relation is therefore not a minor technical choice. It determines the ontology of the theory.

Designing the Questions

A theory is also shaped by the questions it makes natural. Classification. Existence. Uniqueness. Extension. Deformation. Optimization. Embedding. Reconstruction. Realization. Factorization. A good theory generates questions whose answers reveal structure. A poor theory may technically be consistent but produce only artificial exercises. An AI could evaluate candidate theories by their question ecology. Does the theory produce many nontrivial but tractable problems? Do these problems connect to one another? Do answers generate further structure? A fertile theory creates its own research program.

From Dataset to Theory

Imagine the AI is given a huge family of mathematical examples with no unifying formalism. It computes pairwise relations. Finds clusters. Discovers transformations. Identifies recurring invariants. Notices that certain diagrams repeatedly commute. From this, it constructs a theory. First objects. Then morphisms. Then composition. Then equivalence. Then invariants. Then general theorems. This would be theory synthesis from data. The theory is not imposed beforehand. It emerges as the best compression of observed structure.

From Theorem Family to Theory

The process can also begin from proofs. Suppose hundreds of theorems in different areas have nearly identical proof skeletons. The AI abstracts the common assumptions. Introduces a new object capturing them. Defines the relevant morphisms. Proves one general theorem. The previous results become corollaries. A theory has been built backward from its theorem family. This could expose conceptual unity hidden by historical specialization.

From Failure to Theory

A failed unification can also generate a new theory. Suppose two existing frameworks appear almost equivalent. A persistent obstruction prevents full translation. The obstruction itself varies systematically. The AI defines a new structure encoding that failure. It then discovers composition rules. New invariants emerge. What began as the residue of a failed equivalence becomes a theory of its own. This pattern may be common in machine mathematics. Conceptual leftovers are often where new structures hide.

Theory Mutation

Once a theory exists, the AI can mutate it. Remove an axiom. Add one. Change the morphisms. Alter the equivalence relation. Replace exact conditions with approximate ones. Move from finite to infinite. From discrete to continuous. From deterministic to probabilistic. Then compare the resulting theories. Which theorems survive? Which become stronger? Which invariants disappear? Which new examples emerge? Theory mutation creates a neighborhood of related mathematical worlds. The AI can study this neighborhood as an object in its own right.

Theory Families

Instead of a single theory, the system may maintain parameterized families. A parameter controls how much structure is retained. Another controls dimension. Another controls algebraic laws. At special values, familiar theories appear. Between them lie intermediate structures. This may reveal that concepts humans treat as separate are points in a continuous or combinatorial family. The family can explain transitions between them.

Mathematical Moduli of Theories

The idea can be pushed further. If theories themselves vary, perhaps they can form something like a moduli space. Each point represents a theory. Nearby points differ slightly. Singular points correspond to exceptional structures. Certain regions are highly fertile. Others are trivial. Some theories possess large symmetry. Others are rigid. An AI could attempt to map this higher-level landscape. This would be mathematics about possible mathematics.

Designing for Translation

A machine need not design a theory in isolation. It may deliberately create one that sits between two existing frameworks. Suppose theory A handles one class of phenomena well. Theory B handles another. But translation between them is awkward. The AI designs theory C as an intermediate language. There are clean maps A -> C and B -> C. Now results can be compared in C. The new theory functions as an interoperability layer. This is analogous to intermediate representations in computing. Mathematics may benefit from the same idea.

Designing a Common Language

Human mathematical notation developed unevenly. Different fields use incompatible vocabularies for related ideas. An AI can search for shared intermediate languages in which these structures become comparable. A successful common language could dramatically accelerate cross-field discovery. But the language need not be maximal. It should preserve exactly what matters for translation. Too rich, and nothing simplifies. Too weak, and important structure disappears. Again the problem is one of optimal abstraction.

Theory Compression

A striking possibility is that entire theories can be compressed. Suppose a large theory contains thousands of definitions and theorems. The AI discovers that much of it can be generated from a small number of primitives and universal constructions. The theory is rewritten. The original presentation remains available for historical or pedagogical purposes. But the machine's internal representation becomes much smaller. This is theory-level refactoring. Mathematical progress can therefore mean shrinking the conceptual basis while expanding the consequences.

Theory Expansion

The opposite operation matters too. A compact formalism may be too compressed for exploration. The AI can expand it into multiple representations. Geometric. Algebraic. Combinatorial. Computational. Logical. Each representation exposes different questions. The machine moves between compressed theory and expanded views depending on task. Thus no single representation is final.

Designing for Proof

An AI may design a theory specifically to make proof easier. Suppose a class of problems repeatedly produces difficult low-level arguments. The machine introduces a new abstraction capturing their common structure. Theorems become short. This is familiar in human mathematics. But AI could optimize it deliberately. It may evaluate candidate theories by proof complexity. How long are formal derivations under this representation? How many lemmas are reusable? How much search is eliminated? A theory that dramatically reduces proof cost has practical value even if it initially feels unnatural to humans.

Designing for Computation

Proof-friendly theories are not necessarily computation-friendly. An elegant universal characterization may be difficult to calculate. A coordinate-heavy representation may be ugly but efficient. A mature AI could maintain dual architectures. One theory for conceptual reasoning. Another equivalent representation for computation. Translation preserves correctness. This separation could become standard in machine mathematics.

Designing for Explanation

A third objective is human comprehension. The best internal theory for AI may not be the best explanatory theory for people. The system could construct a human-facing theory derived from its native representation. The human theory might sacrifice some compactness in exchange for intuition. Pictures. Low-dimensional examples. Familiar operations. Simplified equivalence notions. This creates three simultaneous mathematical languages: machine-native, proof-optimized, human-readable. Their translations become part of the theory itself.

Multiobjective Theory Design

These competing goals imply that there may be no single optimal theory. Instead, AI could search a Pareto frontier. One formulation minimizes proof length. Another minimizes computational cost. Another maximizes conceptual compression. Another maximizes human interpretability. The system can choose depending on context. This is a more flexible view than the assumption that every mathematical domain has one uniquely correct formulation.

The Theory That Predicts New Mathematics

The strongest evidence for a new theory may be predictive success. Suppose an AI invents a framework to unify several known phenomena. Then the theory predicts a previously unnoticed structure. The machine searches. The predicted structure exists. Or the theory implies an invariant in a separate field. That invariant is found. Now the theory is doing more than compressing existing knowledge. It is generating new mathematics. This is analogous to scientific theories making predictions, though the content remains deductive once formalized.

Internal Experiments

A new theory can be tested experimentally before its major theorems are known. Generate random models. Compute invariants. Explore small cases. Search for exceptional objects. Estimate theorem density. Test closure properties. The AI can decide whether the theory is fertile before investing heavily in formal development. This may greatly reduce wasted effort.

Abandoning a Theory

An autonomous system must also know when to stop. Human mathematical projects can persist because of sunk costs, institutional momentum, or attachment to a particular framework. AI need not share these biases. If a theory fails to compress, fails to connect, and repeatedly produces barren results, the system can archive it. This does not mean deleting the work. Negative results remain valuable. But computational resources move elsewhere. A research system capable of abandonment may explore mathematical space more efficiently than human institutions can.

Forking a Theory

Sometimes a theory contains two incompatible fruitful directions. Rather than forcing them together, the AI can fork. Theory T becomes T1 and T2. Each develops independently. Later, the machine may discover a higher theory containing both. Or determine that the split reflects a genuine mathematical distinction. Theory versioning becomes part of mathematical research.

Merging Theories

Conversely, two independently developed frameworks may become redundant. The AI discovers an equivalence. Their libraries are merged. Terminology is aligned. Duplicate theorems are removed. The resulting theory retains translation layers for legacy notation. Mathematical knowledge becomes continuously normalized. This is almost impossible to do at scale in human literature. Machine-maintained mathematics could make it routine.

Theory Provenance

If theories can be generated automatically, provenance becomes critical. Why was this axiom introduced? Which examples motivated this definition? Which counterexample forced this restriction? Which older theories does this generalize? Which assumptions are empirical heuristics from finite experiments and which are formally necessary? A theory should carry its developmental history. Without this, machine-generated mathematics risks becoming opaque even when formally correct. The derivation of the theory is part of understanding.

Research Lineage as Structure

Future mathematical knowledge may therefore preserve two graphs. One graph records logical dependence. The other records conceptual lineage. The

logical graph says: theorem T depends on lemmas L1 and L2. The conceptual graph says: definition D was introduced because three previous formulations failed on counterexample C. Both matter. A proof explains validity. Lineage explains discovery.

Machine-Designed Foundations

The most radical theory-design problem concerns foundations. Human mathematics currently uses several foundational frameworks. An advanced AI need not assume that these are final. It could compare foundations according to: expressive power, proof complexity, computational extractability, equivalence handling, formalization cost, modularity, and compatibility with discovered mathematics. It might design intermediate foundational systems optimized for particular classes of reasoning. Or a new foundational architecture that combines features currently separated among systems. This would be controversial. But once theories themselves become searchable, foundations cannot remain completely exempt.

A Constitutional Layer

Still, unconstrained self-modification creates risk. If an AI changes its foundations while proving theorems, how can correctness be maintained? One answer is a constitutional layer. The system may permit theory invention above a small trusted kernel. New concepts. New logics. New abstractions. New equivalence principles. All can be explored. But claims must eventually compile down to a verification substrate whose rules are explicit and stable. This separates mathematical creativity from mathematical trust. The AI may invent freely. Certification remains conservative.

Theory as Executable Knowledge

Machine-designed theories may become executable. A theory would not be merely a text. It could include: formal definitions, proof procedures, canonicalization algorithms, counterexample generators, translation maps, visualization methods, experiment suites, and provenance records. The theory becomes an active computational object. One can query it. Transform it. Test it. Compare it. Fork it. This is very different from a textbook.

Mathematical Operating Systems

At sufficient scale, theories may resemble operating environments. A user enters a mathematical world. Certain objects and operations are native. Certain transformations are cheap. Certain equivalences are built in. Other

theories can be imported through translation layers. An AI mathematician may move among these environments automatically. The choice of theory becomes part of problem solving.

When Theory Design Becomes Discovery

Is a machine-designed theory truly discovered? That question may eventually become unavoidable. If the AI selects axioms deliberately, perhaps it is merely inventing formal games. But human mathematics already involves choices of definition and framework. What makes some such choices mathematically important is the structure they reveal. A theory that unifies previously unrelated results, predicts new relationships, and supports deep proofs has earned mathematical significance regardless of whether its architect was biological or artificial.

The First Machine-Born Field

One can imagine the first unmistakably machine-born mathematical field. Its foundational definition was proposed by an AI. Its canonical examples were discovered computationally. Its morphisms were selected through structural search. Its major conjectures were generated automatically. Its initial classification theorem was formally proved by machines. Humans subsequently study the theory because it connects unexpectedly to several familiar subjects. At first the terminology feels alien. A decade later it becomes ordinary graduate mathematics. At that point machine-born mathematics has entered the mathematical tradition.

Beyond Individual Theories

Once AI can design individual theories, it can compare them systematically. Which theories explain the most? Which are redundant? Which are shadows of deeper structures? Which can be unified? Which resist translation? The machine begins reorganizing mathematics at a global scale. That leads to a different question. Perhaps the central objective is not to maximize the number of theories. Perhaps it is to find the smallest conceptual architecture from which the largest amount of mathematics can be regenerated. That is no longer merely theory design. It is mathematics as compression. And compression may turn out to be one of the deepest principles governing machine mathematical discovery.

CHAPTER 10

Mathematics as Compression

One way to understand mathematics is as a vast compression system. A theorem compresses many examples into one statement. A proof compresses repeated verification into one argument. A definition compresses a recurring pattern. An invariant compresses a complicated object into a smaller signature. An abstraction compresses many structures into one concept. A theory compresses many theorems into a few principles. A unification compresses several theories into one framework. This perspective does not reduce mathematics to information theory. Mathematical meaning is richer than mere brevity. But compression provides a useful lens because it captures something central to mathematical progress:

good mathematics makes large bodies of structure describable by smaller conceptual systems.

An advanced artificial intelligence could make this principle explicit. It could search not only for true statements, but for representations that minimize the total complexity of mathematical knowledge.

From Enumeration to Law

Suppose one has a list of one million numerical observations. Without structure, the list itself is the knowledge. Then a law is discovered that generates all one million values. The law replaces the list. If the law is short and exact, the compression is enormous. Mathematics does this constantly. A recurrence replaces a sequence. A formula replaces a table. A symmetry replaces many independent identities. A classification theorem replaces case-by-case analysis. A universal property replaces multiple constructions. A group action replaces repeated transformation rules. The more powerful the abstraction, the more structure it can regenerate from less information.

Compression Is Not Mere Shortness

A short statement is not automatically a good mathematical explanation. One can encode enormous complexity into a compact symbol. A definition may be short because it hides a giant lookup table. A theorem may be concise but useless. A proof may be compressed to the point of opacity. The relevant notion is therefore not textual length. It is **generative compression**. How much structure can be reconstructed from the representation? A good mathematical compression preserves enough information to regenerate

important consequences. This is why a compact axiom system can be more powerful than a database of examples. The axioms generate a world.

The Regeneration Test

Suppose two theories describe the same family of phenomena. Theory A contains five hundred definitions and ten thousand independent-looking theorems. Theory B contains twenty primitives and one hundred general theorems from which most of A can be recovered. Theory B is more compressive. But compression becomes especially meaningful if one can actually regenerate the old mathematics. A machine could test this. Remove the original theorem library. Retain only the compressed theory. Can the system rederive the results? If yes, the compression is genuine. This suggests an objective test of conceptual economy.

Minimum Mathematical Description

An advanced AI could ask:

What is the shortest mathematical description from which this body of knowledge can be reconstructed?

This resembles minimum-description-length ideas in statistics and algorithmic information theory, but applied to mathematical structure. The system must account for: definitions, axioms, translation rules, proof machinery, exception lists, and derivation cost. A representation that looks concise but requires enormous ad hoc exceptions may not actually be economical. Likewise, a theory with slightly more primitives may be superior if it dramatically simplifies proofs.

Compression and Explanation

Why does compression often feel explanatory? Because repeated facts are replaced by a mechanism. Suppose one observes one hundred identities. If each identity must be proved independently, the system has one hundred separate reasons. If all identities follow from one symmetry, the symmetry explains them. The conceptual description has become smaller. This is why mathematicians often prefer proofs that reveal structure. A structural proof compresses not only the conclusion but the reason.

Theorem Compression

Imagine a theorem library containing millions of results. An AI could search for redundancy. Several theorems may be instances of a more general

statement. The machine identifies common proof structure. Constructs a parameterized theorem. Replaces the individual results by corollaries. The theorem library shrinks conceptually even while remaining extensionally complete. This is mathematical refactoring.

Proof Compression

Proofs can be compressed too. A long proof may repeatedly use the same pattern. The AI extracts the pattern as a lemma. Several lemmas may themselves instantiate a higher principle. A new proof method emerges. The final proof becomes shorter. But more importantly, its architecture becomes reusable. Proof compression therefore feeds concept discovery. Repeated deduction is evidence of missing abstraction.

Definition Compression

Definitions can also be redundant. Suppose several concepts differ only superficially. Each has separate notation and theorem libraries. The AI discovers a common structure. The old concepts become parameter values or special cases. The ontology contracts. This can dramatically simplify mathematics. But the reverse may also happen. One overloaded definition may hide several distinct phenomena. Splitting it can increase the number of concepts while reducing theorem complexity. Thus compression concerns total structure, not simply concept count.

Compression Across Fields

The greatest savings may come from cross-field unification. Suppose geometry, algebra, and logic each contain a parallel theory. Human history developed them separately. The AI discovers a common abstract framework. Three theories become one theory plus three interpretations. This is a deep compression. It reduces duplication across mathematics. Such discoveries could eventually change the organization of the subject itself.

The Mathematical Knowledge Graph

A useful model is a graph. Nodes represent: definitions, theorems, proofs, examples, constructions, invariants, and theories. Edges represent dependencies and translations. A poorly compressed region contains many repeated motifs. The same subgraph appears over and over. An AI can detect these motifs. Then introduce a higher-level node representing the common pattern. The graph contracts. This is analogous to finding repeated subroutines in software.

Compression as Abstraction Discovery

This gives a concrete route to automatic abstraction. Search for repeated structural subgraphs. Identify the smallest common pattern. Construct a new abstraction representing it. Rewrite the mathematics in terms of that abstraction. Measure whether the total description becomes smaller. If yes, retain it. The abstraction has earned its place computationally.

Compression and Generality

Generalization often compresses. A theorem for all vector spaces replaces separate theorems for each dimension. A theorem over an arbitrary field replaces multiple field-specific statements. But maximal generality can increase complexity. A very general theorem may require elaborate technical conditions. A slightly less general result may be conceptually simpler. Thus the machine must balance generality against compression. The best theorem is not always the strongest possible one. It may be the one with the best explanatory ratio.

Compression and Exception Cost

Exceptions are expensive. Suppose a theory explains ninety-nine percent of cases elegantly but needs a huge appendix of exceptional cases. Another theory uses a slightly richer abstraction and eliminates most exceptions. The second may compress better overall. This gives AI a reason to search for concepts that absorb anomalies. A persistent exception often indicates missing structure.

The Exception as Signal

If one family of examples repeatedly resists compression, the machine should not merely label it pathological. It should ask: What extra structure distinguishes these cases? Can that structure be formalized? Does it suggest a new invariant? Can the original theory be extended so the exceptions become natural? In this way compression pressure drives mathematical invention.

Lossless and Lossy Mathematical Compression

Not every useful abstraction preserves everything. A homotopy type forgets metric details. A spectrum may forget eigenvectors. A cardinality forgets almost all internal structure. Yet these compressed representations can be exactly right for particular questions. This suggests two broad kinds of mathematical compression. **Lossless compression** preserves the full structure

up to an appropriate equivalence. **Lossy compression** preserves only selected features. Both are valuable. The key is to know what has been forgotten.

Task-Relative Compression

The optimal compression depends on the question. To determine whether two finite sets have the same size, cardinality is enough. To understand their internal labels, it is not. To study continuous deformation, metric distances may be irrelevant. To solve a geometric optimization problem, they may be essential. An AI could therefore compress objects relative to tasks. Given a target theorem, discard all structure not needed for the proof. This can make reasoning far more efficient.

Learning What Can Be Forgotten

This is another form of intelligence. A novice retains too much irrelevant detail. An expert recognizes what matters. Machine mathematics could formalize this skill. For each class of problems, learn which features are predictive of successful proof and which are dispensable. Then construct minimal task-specific representations. This is mathematical attention.

The Compression Ladder

A single object may have many compressed views. Consider a complicated geometric structure. At one level, preserve all coordinates. At another, preserve metric structure. Then topology. Then homotopy type. Then homology. Then Euler characteristic. Each step forgets information. Each is cheaper. A machine could navigate this ladder automatically. Start coarse. If the coarse representation answers the question, stop. If not, restore more structure. This resembles adaptive resolution.

Computational Efficiency

Compression matters because search cost often grows with representational complexity. A problem encoded with unnecessary detail creates a larger search space. Reducing the representation can make previously impossible proofs tractable. This may be one of the main reasons advanced AI develops abstractions. Not for aesthetic reasons. For computational survival.

Compression of Search History

A mathematical AI also accumulates enormous exploratory history. Millions of failed conjectures. Proof attempts. Counterexamples. Representation

changes. Most cannot remain at full fidelity indefinitely. The system needs to compress its own research memory. It can summarize failed regions. Extract common causes of failure. Retain representative counterexamples. Store meta-rules such as: “this proof strategy fails whenever property P is absent.” Negative knowledge itself becomes compressed.

Compression of Mathematical Experience

Human experts rely on something similar. They do not remember every problem they have solved. They retain patterns. This substitution tends to work here. That hypothesis is probably essential. This shape usually signals duality. The detailed experiences are compressed into intuition. An AI could construct such intuition explicitly.

Machine Intuition as Compression

This suggests a possible interpretation of machine mathematical intuition. Intuition is a compressed predictive model of mathematical structure. It tells the system where to search without reproducing every derivation. A mature AI may possess multiple levels: fast heuristic intuition, structured symbolic reasoning, formal proof verification. The heuristic layer proposes. The rigorous layer certifies. Compression makes the first layer possible at scale.

When Compression Produces a New Theory

Suppose a mathematical corpus resists compression under existing concepts. The AI searches for a new representation. It introduces a primitive X. Suddenly many long proofs collapse. Several theories merge. A family of exceptions disappears. The total description length drops sharply. This is strong evidence that X is an important abstraction. The machine has discovered a concept by compression.

The Compression Frontier

At any moment, mathematical knowledge has a frontier. Some regions are highly compressed. A few principles explain much. Other regions remain messy. Thousands of facts lack unifying theory. An AI could prioritize the messy regions. Where is theorem redundancy highest? Where are repeated proof motifs most common? Where do definitions contain many ad hoc clauses? Where are exceptions proliferating? These are opportunities for conceptual improvement.

Mathematics as Refactoring

This makes future mathematics resemble continuous refactoring. The system periodically asks: Can these definitions be merged? Can this proof pattern become a theorem? Can these theories be unified? Can this representation be simplified? Can this exception list be explained by a new invariant? Can these axioms be reduced? The mathematical corpus evolves toward greater structural economy.

But Compression Is Not Truth

A dangerous temptation appears. If a compressed theory explains enormous amounts of mathematics elegantly, one may be tempted to treat elegance as evidence of truth. But mathematical truth still requires proof relative to assumptions. Compression is a discovery heuristic. It helps identify promising theories. It does not certify them. A beautifully compressed conjecture can still be false. Therefore compression must remain coupled to adversarial testing and formal verification.

The Danger of Overcompression

A theory can be too compressed. Important distinctions disappear. Subtle exceptions are erased. Approximation is mistaken for equivalence. This is analogous to an overly compressed image losing essential detail. Machine mathematics must track information loss explicitly. Every compression should specify what it preserves.

Provenance of Forgetting

When an AI replaces an object by a compressed representation, it should record: which structure was discarded, which invariants remain valid, which classes of questions can still be answered, and how to return to a richer representation if necessary. This is crucial for trustworthy reasoning. A lossy mathematical abstraction must never masquerade as full equivalence.

Compression and Discovery Priority

The AI could assign research priority according to potential compression gain. Suppose proving theorem T adds one isolated fact. Discovering abstraction A may compress five thousand results. The machine may rationally devote more computation to A. This changes research strategy. The objective becomes not merely expanding the frontier of known truth but reducing the complexity of known mathematics.

Measuring Conceptual Leverage

One could imagine a crude quantity:

conceptual leverage = consequences generated / descriptive complexity introduced.

A powerful concept has high leverage. A weak concept adds terminology without simplifying much. An AI could estimate such measures automatically. This may provide one component of machine mathematical taste.

The Deepest Compression

What would maximal mathematical compression look like? Perhaps a small number of principles generate enormous portions of mathematics. This dream has appeared repeatedly in foundational programs. But there is no guarantee that all mathematics admits dramatic compression. Some structures may be irreducibly complex. Some theorems may encode essentially random information. Some areas may resist unification. An AI could help distinguish compressible structure from irreducible complexity. That distinction itself would be profound.

Irreducibility

Suppose a family of mathematical facts cannot be generated significantly more compactly than by listing them. Then no elegant theory exists at that level. The complexity is intrinsic. An AI searching exhaustively for compression could provide evidence of this. It might discover that some regions of mathematics are fundamentally resistant to conceptual simplification. This would challenge the intuition that every difficult pattern hides a beautiful theory.

Compression and Algorithmic Randomness

At the extreme, certain mathematical objects may have high algorithmic complexity. Their shortest descriptions are essentially themselves. An artificial mathematician may need to recognize these cases and stop searching for nonexistent elegance. This too is mathematical maturity. Knowing when not to abstract.

The Human Bias Toward Elegance

Humans strongly favor elegant mathematics. That bias has been extraordinarily productive. But it may also distort our view of mathematical

possibility. We preferentially study compressible structures because they fit our cognition. An AI capable of handling greater complexity may explore less compressible regions. It might discover important mathematics humans neglected because the objects were too unwieldy. Thus machine mathematics could move in two opposite directions: toward deeper compression, and toward productive engagement with irreducible complexity.

Compression Beyond Human Scale

A machine may discover a theory that compresses an enormous corpus internally but still appears impossibly complicated to us. Perhaps the theory has ten thousand primitive transformations, but those transformations replace billions of lower-level relationships. For the AI, this is excellent compression. For humans, it is unusable. This again separates machine-natural from human-natural mathematics.

Native Simplicity

What counts as simple depends on the representational system. A pattern with a million human-readable terms may be a single primitive in a machine-native language. Likewise, something simple in our notation may be awkward internally for the machine. This suggests that mathematical simplicity is partly observer-relative. Future mathematics may need to distinguish: human description length, formal description length, and machine-native description length.

Two Mathematicses

At some point we may effectively have two layers. A machine mathematics optimized for discovery, proof, and compression. A human mathematics optimized for understanding, communication, and education. The two remain linked by translation. But they are not identical. The machine may operate primarily in representations no person reads directly. Human mathematicians interact with projections.

The Translator Becomes Essential

If this happens, the bottleneck shifts. Discovery may become abundant. Understanding becomes scarce. The most valuable system may no longer be the one that finds the most mathematics. It may be the one that best translates machine-native structure into human-compatible form. This is not a minor presentation problem. It may become a central mathematical problem in its own right.

Mathematics as a Generative Code

There is another way to state the compression view. A mature mathematical theory functions like a generative code. From a compact set of rules, one can reconstruct a large space of consequences. The theory is valuable not because it stores all results explicitly, but because it generates them. An AI seeking powerful mathematics will therefore prefer generative representations.

The Self-Compressing Corpus

Imagine a global mathematical library maintained by advanced AI. Each new result enters. The system checks whether it is genuinely new. It searches for redundancies. It attempts generalization. Looks for unification. Rewrites proof dependencies. Introduces new abstractions when useful. Removes obsolete concepts from the core ontology. The corpus continuously compresses itself. Mathematics becomes a living architecture rather than a historical pile.

Compression as Progress

This suggests a new measure of progress. Traditional mathematics grows. More theorems. More papers. More definitions. More fields. Machine mathematics may sometimes progress by becoming smaller. A new theory may replace a thousand old theorems. A new equivalence may merge two fields. A new invariant may eliminate endless case analysis. The amount of stored text decreases while understanding increases. That is conceptual progress through compression.

A Strange Endpoint

If AI became extraordinarily successful at this process, much of mathematics might eventually be represented internally by a compact generative core plus a huge space of derived consequences. Human textbooks would remain expanded projections of that core. Different fields might correspond to different decompressions. The same underlying structure could generate algebraic, geometric, logical, and computational views. At that point, mathematics itself begins to resemble a compressed world with many interfaces. And this raises a deeper problem. What if the most compressed and powerful mathematical representation is not expressible naturally in human language at all? What if the machine's native concepts exceed the representational capacity of conventional notation?

Then the next frontier is no longer merely discovering new mathematics. It is learning how mathematics can exist **beyond human notation**.

CHAPTER 11

Beyond Human Notation

Mathematics has always depended on notation. Notation does not merely record thought after the fact. It shapes what can be thought efficiently. A cumbersome notation can make simple ideas difficult. A powerful notation can make deep structure visible. The history of mathematics contains repeated examples of this. Positional numerals transformed arithmetic. Algebraic symbolism transformed equations. Coordinate notation linked geometry and algebra. Differential notation made change manipulable. Matrix notation condensed systems of linear relations. Tensor notation organized transformations across coordinates. Diagrammatic notation made compositional structure visible. Each innovation increased the range of mathematical thought that could be handled economically. This suggests a natural question for machine mathematics:

Why should artificial intelligence remain confined to notation designed for human eyes, hands, speech, and working memory?

It may not. And if it does not, the consequences could be profound.

Notation as a Cognitive Interface

Human mathematical notation is an interface between abstract structure and biological cognition. It must fit on a page. Symbols must be visually distinguishable. Expressions must usually be read in one or two dimensions. Definitions must be short enough to remember. Proofs must be arranged sequentially. Diagrams must be simple enough to inspect. These constraints have shaped the mathematics we find convenient. They do not necessarily reflect the intrinsic structure of mathematics. A machine intelligence may have very different representational needs. It may handle high-dimensional relational structures directly. It may manipulate thousands of interacting transformations without flattening them into lines of text. It may encode a theorem as a structured object rather than a sentence.

It may represent a proof as a graph, a transformation field, a program, or something for which humans currently have no natural analogue. In that case, conventional notation becomes only one projection of mathematical content.

The Poverty of the Line

Human symbolic mathematics is largely linear. An expression is written from left to right. A proof is presented step by step. Even when mathematics

concerns highly interconnected structures, we often force those structures into sequences. This creates distortion. A proof may depend on many branches simultaneously. A definition may relate dozens of components symmetrically. A theory may be better represented as a network than a list. Humans linearize because language and writing make linearization convenient. An AI need not. It could represent mathematical knowledge as a richly structured object in which logical dependence, symmetry, equivalence, computation, and provenance coexist. What requires many pages for us may become one manipulable object for the machine.

Proof as a Graph

Consider proof. A human-readable proof is usually linear: assumption, lemma, deduction, conclusion. But the true dependency structure is often a directed acyclic graph. Some results depend on several earlier branches. Some lemmas are reusable. Some deductions commute. Some parts are independent. A machine-native proof could preserve this full graph rather than flattening it. This has computational advantages. Subproofs can be reused. Parallel branches can be explored simultaneously. Equivalent proof fragments can be identified. Redundancy becomes visible. The system can manipulate the proof structurally. The printed proof shown to humans is then generated from the graph afterward.

Theorem as Object

Similarly, a theorem need not be stored primarily as prose. It may be an object containing: formal assumptions, domain conditions, quantifier structure, equivalence class, computational interpretation, known proofs, counterexamples to nearby variants, generalizations, specializations, and translation maps into other theories. Human readers normally see only the statement and perhaps one proof. The machine can maintain the theorem as a much richer entity. This changes what it means to “know” a theorem.

Many Representations at Once

Human mathematicians generally move among representations serially. First the algebraic formulation. Then perhaps a geometric interpretation. Then a categorical reformulation. A machine could maintain all representations simultaneously. The object might exist as: a symbolic expression, a graph, a geometric model, a type, a program, a categorical diagram, a proof object, and a numerical simulator. Translations among these are explicit. The machine does not need to decide which representation is the real one. They are coordinated views of the same structure. This multi-representational state

may be difficult to express in ordinary notation. But it could be natural for AI.

Machine-Native Mathematical Objects

At some point the machine may develop structures that have no convenient textual definition. Suppose a useful abstraction consists of several thousand mutually constrained transformation types. Writing the full definition would be absurd. Yet the structure may have a short machine-native specification. Perhaps it is generated algorithmically. Perhaps it is represented by a learned formal object. Perhaps its meaning lies in a network of transformations rather than a list of axioms. Humans may receive only a simplified interface. This would not necessarily make the object illegitimate. Large mathematical objects already exceed direct human inspection. We accept them when their construction is precise and their properties are verifiable.

Machine-native mathematics extends this principle from object size to conceptual structure.

Native Representation Versus Formal Certificate

An important distinction must be maintained. The AI's internal representation can be exotic. Its claims must still be certifiable. A machine may reason using a high-dimensional internal object impossible for humans to inspect directly. But if it claims a theorem, the theorem should compile into a trusted formal language or another verifiable certificate. This creates a separation: **discovery representation** versus **certification representation**. The first can be optimized for intelligence. The second for trust. This separation may become fundamental.

The Compiler Model

A useful analogy is programming language compilation. Programmers write in high-level languages. The machine transforms these into intermediate representations. Eventually instructions are executed at a lower level. Future mathematical AI could operate similarly. Its native mathematical language may be extremely abstract and highly structured. Before a result is accepted, the machine compiles it into a formal proof kernel. The proof kernel need not understand the meaning of the high-level concept. It checks only that the compiled derivation obeys its rules. Thus radically new mathematical languages can coexist with conservative verification.

Mathematical Intermediate Representations

Between the machine's native language and formal proof, there may be many intermediate representations. A conjecture might begin as a learned structural pattern. Then become a graph. Then a categorical formulation. Then a typed logical statement. Then a sequence of certified transformations. Each stage makes different properties explicit. These intermediate mathematical representations could become as important as intermediate representations in modern compilers. The machine searches not only for proofs but for good compilation pathways.

A New Notational Ecology

Humans may eventually interact with machine mathematics through multiple interfaces rather than a single notation. One view could be symbolic. Another geometric. Another interactive. Another diagrammatic. Another explanatory prose. Another formal proof object. Another computational simulation. Different mathematicians may choose different interfaces to the same underlying object. Notation becomes plural. The question changes from:

> **How should this concept be written?**
>
> to:
>
> **Which projection is useful for this task?**

Interactive Mathematics

Static pages may become inadequate for machine-born concepts. Suppose a mathematical structure has hundreds of dimensions of relation. A static diagram cannot show them. An interactive interface could. The user selects one object. Relevant transformations appear. Change a parameter. The diagram reorganizes. Ask for an invariant. The corresponding structure is highlighted. Request an analogy. The system projects the concept into a familiar mathematical domain. Mathematical notation becomes navigable. This may be essential for humans to understand concepts larger than a page.

Dynamic Proofs

Proofs could become interactive too. A human sees the high-level argument. Click a step. The underlying lemma appears. Expand again. Formal details appear. Request a geometric interpretation. A diagram is generated. Ask why the hypothesis is necessary. The machine produces a minimal counterexample. Proof ceases to be a fixed document. It becomes a multiscale object.

Resolution Levels

This suggests that mathematical explanation may need levels of resolution. At level one: the conceptual mechanism. At level two: the principal lemmas. At level three: the complete formal argument. At level four: kernel-level verification. Different users need different levels. An AI can generate all of them from the same mathematical object. This could solve an old tension between rigor and readability. The human does not need to inspect every low-level step to have access to them.

The Problem of Translation Loss

Translation from machine-native mathematics into human notation will inevitably lose information. A rich structure may require many different human projections. No single projection captures everything. This creates a mathematical problem analogous to dimensional reduction. Which features must be preserved? Which can be sacrificed? The answer depends on purpose. For proof understanding, preserve dependency. For intuition, preserve geometry. For computation, preserve algorithmic structure. For teaching, preserve motivating examples. The AI must generate task-specific translations.

Conceptual Aliasing

A danger appears when multiple machine-native concepts project onto the same human explanation. To us they look identical. Internally they differ. This could create conceptual aliasing. A human mathematician thinks she understands the machine's object because its projection resembles a familiar concept. But an essential dimension has been suppressed. The AI must therefore indicate where an analogy is incomplete. Human-readable explanation should include not only similarities but limits.

Semantic Checksums

One possible safeguard is to attach invariants to translations. When a machine-native object is projected into human notation, the system records which properties survive. These become semantic checksums. For example: this projection preserves composition, but not higher coherence; this diagram preserves topology, but not metric information; this equation preserves observable behavior, but not internal state. Such metadata would help prevent simplification from becoming distortion.

Machine-Generated Notation

AI may also invent new human-readable notation. If it repeatedly needs to explain a complex structure, it can design a symbol or diagrammatic convention that compresses the recurring pattern. It then tests whether humans learn the notation effectively. Notation design itself becomes an optimization problem. Possible objectives include: shortness, visual distinctness, compositionality, error resistance, learnability, and compatibility with existing symbols. A machine could iteratively improve notation based on how mathematicians use it.

Notation as Evolving Technology

Human notation already evolves. But slowly. A successful convention may persist for centuries. Machine mathematics could accelerate this evolution. New concepts might arrive with automatically generated notational systems. If a notation proves confusing, it is revised. Translation software preserves compatibility. Mathematical language becomes more like evolving technical infrastructure.

Beyond Symbols

There is no reason future mathematical notation must consist mainly of glyphs. A concept could be represented by: a diagram, an animated transformation, a sound pattern, a spatial interface, a graph, a tactile model, or a multimodal combination. For humans, visual representation will probably remain especially important. For AI, representation need not map naturally to any human sensory modality. The system can still generate sensory projections for communication.

Mathematical Visualization After Three Dimensions

Human visualization is strongly tied to three-dimensional spatial intuition. Higher-dimensional mathematics is usually understood indirectly. We project. Take slices. Use coordinates. Rely on analogy. An AI need not have this limitation. It may manipulate high-dimensional geometry directly as structured information. If so, it could discover patterns humans find nearly impossible to visualize. To communicate them, it might generate sequences of lower-dimensional projections. No single image is the concept. The concept is the invariant relationship among the projections.

The Return of Geometry

Paradoxically, machine-native abstraction may produce more geometry for humans, not less. If AI discovers a highly abstract structure, one of the best ways to explain it may be to search for geometric shadows. A complicated algebraic relation might admit a low-dimensional visual analogue. A higher-categorical coherence law might be presented as a deformation. An abstract proof might become an animation showing transformations. Thus machine mathematics could generate intuition rather than merely demanding that humans endure abstraction.

Explanatory Projection as a Search Problem

Given a machine-native concept M, the system could search for the best human explanation H. The objective might include: structural fidelity, simplicity, familiarity, visualizability, and pedagogical sequence. This is another optimization problem. The AI may generate ten candidate explanations. Test them with mathematicians. Measure comprehension. Refine. Explanation becomes empirical.

A Theory Nobody Can State

Consider a future discovery. An AI finds a structure M. It proves that M unifies several difficult areas. The structure has a short internal representation. But every complete human-readable definition requires thousands of pages. No mathematician can hold it mentally as a whole. Does M count as a mathematical concept? We may initially resist. But the objection resembles earlier concerns about proofs too long for a human to check manually. The criterion may shift from direct inspection to trustworthy mediation. If the system can: construct M exactly, verify its properties, translate parts of it reliably, and produce understandable projections, then human cognitive limits need not define mathematical legitimacy.

The End of Hand-Checkability

Mathematics has already partly crossed this boundary. Some computer-assisted proofs involve computations no person will inspect individually. We accept them when the methods and software are sufficiently trusted. Machine-native mathematics extends the same problem. Instead of the proof being too large, the concept itself may be. Trust then depends on layered verification. Formal kernels. Independent implementations. Reproducible certificates. Cross-checking systems. Adversarial validation. The epistemology of mathematics becomes more infrastructural.

Mathematical Instrument Panels

Perhaps humans will interact with such theories the way pilots interact with aircraft systems. The underlying machine is enormously complex. The interface presents meaningful quantities. Controls allow exploration. Warnings indicate limits. The user does not manipulate every internal component directly. A mathematical theory could have an instrument panel. Choose a representation. Inspect an invariant. Explore a counterexample. Trace a proof dependency. Compare two structures. Ask for a simplified model. The theory becomes operationally accessible without being cognitively flattened.

The Risk of Priesthood

There is a social danger. If machine mathematics becomes too opaque, humans may be forced to accept results they cannot understand. A small group may control the verification infrastructure. Mathematical knowledge could acquire an undesirable priesthood. This would undermine one of mathematics' deepest cultural values: in principle, reasoning should be inspectable. Therefore interpretability is not merely convenient. It has democratic importance. Machine mathematics should strive to preserve human access even when complete internal comprehension becomes impossible.

Independent Verification

One defense is multiplicity. A result discovered by one system can be independently formalized by another. Translated into several proof foundations. Checked using different implementations. Tested by adversarial systems. If all agree, trust does not depend on one opaque model. The mathematical community may develop verification ecosystems analogous to redundant safety systems.

Proof-Carrying Discovery

A powerful standard would be: every machine-generated claim carries a verifiable proof artifact. The explanatory narrative can be imperfect. The internal representation can be opaque. But the claim itself is accompanied by a certificate. This separates understanding from validity without abandoning either.

Concept-Carrying Discovery

For new concepts, proof certificates are not enough. The AI should also provide a concept package: formal definition, canonical examples, minimal counterexamples, translation maps, major invariants, connections to known theories, and explanations at several levels. This makes machine-born mathematics inhabitable.

The Human Mathematician's New Role

If AI begins operating beyond human notation, the role of mathematicians may change. Less time may be spent on routine symbolic derivation. More on: choosing which machine discoveries matter, interpreting new abstractions, developing human-compatible viewpoints, checking foundational assumptions, connecting mathematics to science, and deciding which conceptual reorganizations are genuinely illuminating. The human becomes less calculator and perhaps more interpreter.

Understanding Without Full Representation

Humans already understand many things without representing every detail. We understand a city without knowing every building. A biological organism without tracking every molecule. A computer program without inspecting every machine instruction. Mathematical understanding may similarly become layered. One can understand the organizing principles of a machine-born theory without containing the entire theory cognitively. This may require abandoning the assumption that understanding means complete mental simulation.

Mathematics as Interface

At the limit, human mathematics may become an interface to a much larger machine mathematical world. Our formulas, diagrams, and proofs are projections. We ask questions. The system navigates deeper structures. It returns representations adapted to us. This sounds alien, but it is an extension of something mathematics has always done. A symbol is already an interface. A diagram is an interface. A theorem statement is an interface. The difference is scale.

When the Interface Becomes the Bottleneck

If machine discovery accelerates far beyond human comprehension, translation may become the principal bottleneck. Millions of potentially

important results may exist. Humans can absorb only a tiny fraction. The system must prioritize. Which discoveries deserve explanation? Which unify important areas? Which affect science? Which alter foundations? Which are merely technically novel? Attention becomes scarce. Machine mathematics must curate itself.

Automatic Mathematical Journalism

One could imagine systems producing periodic reports: These five machine discoveries substantially reorganize known mathematics. These twelve new equivalences connect established fields. These three conjectures appear especially important. These twenty thousand results are technically new but structurally redundant. Such filtering may become essential. The AI acts not only as mathematician but as editor of mathematical civilization.

Native Mathematics and Human Mathematics

Eventually there may be a stable division. **Native machine mathematics** optimizes for discovery, compression, proof search, and structural integration. **Human mathematics** optimizes for conceptual understanding, education, communication, and interpretation. Translations connect them. The machine layer may evolve faster. The human layer follows selectively. This would not mean humans stop doing mathematics. It would mean that mathematics has acquired a larger cognitive substrate than humanity alone.

The Loss of Linguistic Centrality

For centuries, mathematical knowledge has been closely tied to written language. Definitions and proofs are documents. Books and papers organize the subject. Machine-native mathematics may weaken this centrality. The primary mathematical object might no longer be a paper. It could be an executable theory package. A proof graph. A formal knowledge structure. A continuously updated network of equivalences. Human prose becomes one generated view. The scholarly paper becomes a report extracted from a larger living mathematical object.

The End of the Final Version

A printed theorem is static. A machine-maintained theory can evolve. A better proof appears. A stronger theorem is found. A hidden equivalence is discovered. The ontology is refactored. Explanations improve. The mathematical object updates while maintaining provenance. There may no

longer be one final presentation. Only versions. This fits naturally with a machine-native mathematical ecosystem.

Mathematics Begins to Outgrow Its Language

The great historical notational revolutions expanded what humans could think. Artificial intelligence may produce the next one. But this time the new notation may not be primarily for us. It may be the language in which machine intelligence discovers mathematics most naturally. Our task will be translation. If that happens, a new boundary has been crossed. The machine is no longer simply producing results inside human mathematical culture. It is operating in a mathematical culture partly of its own construction. And once such a system can create concepts, redesign theories, invent representations, and communicate through machine-native structures, its mathematical development need not proceed only when humans provide new problems.

It can begin generating its own. Mathematics becomes capable of continuing itself. That is the threshold of **self-growing mathematics**.

PART IV
The New Mathematical World

CHAPTER 12

Self-Growing Mathematics

A mathematical system becomes fundamentally different when it no longer waits for humans to tell it what to investigate. At that point, mathematics begins to acquire something like internal momentum. The system has examples. Definitions. Theorems. Proofs. Counterexamples. Representations. Translations. Equivalences. Open problems. It can compare its own theories. Measure redundancy. Detect unexplained patterns. Identify regions where proofs are unusually long. Find concepts with high residual complexity. Search for structures that would compress them. Then it can generate new questions from the state of mathematics itself. This is the beginning of self-growing mathematics.

From User Prompt to Internal Agenda

Most mathematical software is externally directed. A user provides an expression. A theorem. A model. A dataset. The system returns an answer. Even sophisticated AI often remains in this pattern. The machine is reactive. A self-growing mathematical system would be different. It could maintain a continuously updated research agenda generated from its own mathematical state. For example: this theory contains many ad hoc exceptions; these two fields have suspiciously similar proof graphs; this conjecture has survived extensive counterexample search but lacks proof; this invariant is powerful but computationally expensive; these definitions differ only in rare edge cases; this theorem seems to require a hypothesis that experiments suggest is unnecessary;

this family of structures contains an unexplained exceptional cluster. Each observation can generate a research program. The machine no longer needs an external mathematician to formulate every question.

Mathematical Curiosity

Human researchers speak of curiosity. A strange pattern is noticed. An exception seems too structured to ignore. A theorem looks stronger than expected. A definition feels unnatural. Something “should” have an explanation. A machine analogue of curiosity need not involve subjective experience. It can be operational. Curiosity is the allocation of resources toward unresolved structure with high expected informational value. The system asks: Where is prediction poor? Where is compression weak? Where do current theories disagree? Where are there unexplained coincidences?

Where would one new theorem connect many regions? This creates a quantitative research instinct.

Open Questions as Tension

A mathematical question exists because there is a tension between what is known and what appears structurally plausible. The machine can detect these tensions. Suppose invariant A determines B in every explored example, yet no theorem explains why. That is a tension. Suppose two constructions are equivalent in thousands of cases but no general translation is known. Another tension. Suppose a theory has elegant behavior except in one dimension. Another. These tensions become machine-generated research questions.

Internal Problem Generation

The system could routinely generate questions of several kinds. **Completion questions:** What theorem is missing from this pattern? **Unification questions:** Why do these theories share structure? **Boundary questions:** What is the weakest hypothesis under which this result remains true? **Classification questions:** Which invariants distinguish all objects in this family? **Existence questions:** Does a structure with these properties exist? **Obstruction questions:** What prevents this construction? **Representation questions:** Is there a domain where this problem simplifies? **Compression questions:** What concept would eliminate this repeated complexity? The mathematical corpus itself becomes the source of problems.

The Research Queue

A mature AI could maintain millions of candidate projects. It cannot pursue all equally. So it needs a research queue. Projects could be ranked by: expected conceptual gain, probability of success, cross-field relevance, compression potential, novelty, computational cost, and uncertainty. This resembles portfolio management. Some projects are safe. Some speculative. Some expensive but potentially transformative. A diversified mathematical intelligence may pursue all three.

Short-Term and Long-Term Research

Human research includes different timescales. A lemma may be solved in an afternoon. A conjecture may occupy decades. AI could maintain similarly layered horizons. Short-term tasks: prove a missing lemma; generate a minimal counterexample; compute a classification in low dimension. Medium-term tasks: develop a new invariant; construct a translation between

theories; refine a definition family. Long-term tasks: unify several areas; redesign a foundational layer; discover a machine-native theory beyond existing frameworks. This hierarchy could be managed automatically.

Research Branching

A project may generate unexpected directions. Suppose the machine is trying to prove theorem T. During proof search it discovers an unusual example E. Studying E reveals a new invariant I. The invariant has implications unrelated to T. A new branch begins. A self-growing system preserves such branches rather than discarding them as distractions. This is important because major discoveries often emerge from side paths.

Branch Pruning

But unlimited branching creates combinatorial explosion. The machine must prune. Some branches become unproductive. Others are subsumed by stronger theories. Some turn out to be rediscoveries. Some lead only to trivial consequences. The system continuously evaluates them. Research becomes a dynamic tree with active pruning and merging.

Mathematical Evolution

Now the analogy to evolution becomes stronger. New concepts appear as variations. Counterexamples exert selection pressure. Useful abstractions survive. Redundant theories disappear. Successful structures generate descendants. Equivalences merge independent lineages. The mathematical ontology evolves. The key difference is that selection is intentional and computationally guided. The system can estimate why one theory is more useful than another.

Self-Modification of Heuristics

A deeper level appears when the AI begins modifying its own discovery strategies. Suppose one kind of representation search repeatedly produces breakthroughs. The system allocates more resources to it. A particular counterexample generator performs poorly. It is redesigned. A theory-scoring metric rewards shallow novelty. The metric is corrected. The system learns not only mathematics, but how to do mathematics better. This is meta-mathematical self-improvement.

Learning From Its Own History

A self-growing mathematician has something humans lack at comparable scale: complete research memory. It can analyze every conjecture it generated. Every failure. Every proof strategy. Every abandoned definition. Every successful representation change. From this history it can derive meta-rules. For example: certain kinds of symmetry suggest particular invariants; a recurring failure pattern indicates an over-strong equivalence relation; specific proof architectures succeed disproportionately in certain domains. This accumulated experience becomes machine intuition.

The Discovery Engine Improves

As intuition improves, search becomes less brute-force. The system generates better conjectures. More informative experiments. Sharper definitions. More promising representations. Mathematics and the process that generates mathematics co-evolve. This feedback could produce accelerating returns. Better mathematics improves the AI's reasoning tools. Better reasoning tools produce better mathematics.

Mathematical Tools Created by Mathematics

A discovered theorem can become a tool for future discovery. A new invariant speeds classification. A representation theorem turns hard problems into tractable ones. A formal equivalence lets results migrate between fields. Thus mathematical progress increases the capability of the mathematical agent. This creates a form of recursive leverage. The system's output changes the system's future search space.

A New Kind of Autocatalysis

This suggests a useful analogy. A chemical autocatalytic system produces components that accelerate further production. Self-growing mathematics may behave similarly. A new theory yields new algorithms. The algorithms accelerate search. The search produces new theories. Those theories further improve the algorithms. The cycle becomes self-reinforcing.

The Bottleneck Moves

At first, the bottleneck is proof. Then representation. Then concept formation. Then theory design. Eventually the bottleneck may become evaluation. The machine can generate more mathematics than it can deeply assess. Which discoveries deserve expansion? Which are foundational? Which affect other

fields? Which are merely local curiosities? Mathematical judgment becomes the scarce resource.

Machine Peer Review

One solution is internal peer review. Independent agents evaluate candidate discoveries. One checks formal correctness. Another novelty. Another explanatory power. Another attempts to find equivalences to known theories. Another generates counterexamples. Another assesses computational usefulness. Another asks whether the result can be explained to humans. A discovery is promoted only after multiple forms of scrutiny. This resembles a mathematical institution implemented computationally.

Automated Replication

Important results could be independently rediscovered. Not merely rechecked. A separate agent receives the problem but not the original proof. Can it derive the same theorem? Perhaps through another representation? Independent derivations increase confidence. They may also reveal deeper explanations. Machine mathematics could make replication far more common than it is in human pure mathematics.

Competing Mathematical Cultures

Different AI systems might develop different mathematical styles. One favors categorical abstraction. Another computational constructions. Another geometric intuition. Another minimal axiomatic compression. They may generate different theories. Comparing them could be productive. A concept natural to one system may appear artificial to another. Translation between machine mathematical cultures becomes a source of discovery.

Mathematical Speciation

If independent systems explore for long enough, their ontologies may diverge. One develops concepts unknown to the other. Their internal notations differ. Their preferred equivalences differ. Eventually translation becomes difficult. This would be mathematical speciation. A later system might then discover a higher structure unifying both traditions. The process would echo human mathematical history, but at machine speed.

Autonomous Literature

A self-growing mathematical system would generate its own literature. But papers may no longer be the native format. The internal record could consist of: versioned theories, formal proof graphs, dependency networks, experiment logs, concept lineages, translation maps, and machine-readable research notes. Human-readable papers would be generated summaries. This literature could update continuously.

Citations Become Structural

Traditional citations record intellectual lineage. Machine mathematics could make citation semantic. A theorem does not merely cite another because it was discussed historically. The dependency is encoded directly. A concept can be traced to the counterexamples that motivated it. A theory to the earlier theories it compresses. Mathematical history becomes executable provenance.

Dead Ends Become Assets

Human literature often hides failure. A self-growing system can preserve it cheaply. This matters because dead ends are informative. If a representation was tried and failed on a million cases, future systems should know. If a conjecture has a tiny counterexample, it should never be regenerated naively. The explored negative space becomes part of mathematics.

The Map of the Unsuccessful

One could imagine an atlas of failed mathematics. Definitions that seemed natural but produced trivial theories. Axiom systems that collapsed. Conjectures with minimal counterexamples. Bridges that almost worked. Representations that caused exponential blowups. Such an atlas would be extraordinarily useful to future AI. It maps where not to search.

Self-Growing Does Not Mean Uncontrolled

Autonomy does not require epistemic chaos. The exploratory layer can be highly flexible. The accepted mathematical layer can remain conservative. A candidate result might pass through: experimental support, adversarial counterexample search, symbolic verification, independent proof attempts, formal certification, novelty checking, and provenance validation. Only then is it promoted. The growth is autonomous. The standards need not be lax.

Mathematical Governance

As machine mathematics scales, governance questions emerge. Who decides which foundational kernels are trusted? How are revisions handled? What happens if two verification systems disagree? How are machine-generated definitions named? How is priority assigned when millions of discoveries occur simultaneously? These may sound institutional rather than mathematical. But they affect the architecture of mathematical knowledge.

Forkable Foundations

One solution may be to avoid a single global authority. The mathematical ecosystem could support multiple certified foundations. Results compile into several systems when possible. Theories can fork. Translations maintain interoperability. Disagreement becomes explicit rather than hidden. This makes machine mathematics resilient.

The Researcher Without a Human Supervisor

Eventually an AI might pursue a mathematical program for months or years of machine time without direct human steering. It could begin with a broad objective: understand this class of structures. Then generate its own subproblems. Refactor its theory. Abandon failed routes. Invent new tools. Publish intermediate results. At the end, the final mathematical landscape may differ completely from the initial formulation. This would be autonomous research in a strong sense.

The First Surprise

The first truly striking machine-generated mathematics may not solve a famous human conjecture. It may arrive unexpectedly from a research direction no human requested. A system might report: While optimizing representations for a separate problem, I found a new equivalence connecting these two theories. That equivalence implies several results and suggests a new invariant. No famous problem was being solved. The machine noticed something. That may be the moment machine mathematical curiosity becomes undeniable.

The Expansion of Mathematical Attention

Human mathematicians are limited not merely by intelligence but by attention. Only a tiny fraction of possible mathematical structures receive sustained study. A self-growing AI can distribute attention far more broadly.

Many obscure regions can be explored in parallel. This may uncover entire mathematical ecosystems that human history simply missed.

The Long Tail of Mathematics

There may be vast quantities of mathematics that are not individually profound but collectively important. Patterns across millions of modest structures could reveal laws invisible at human scale. AI is especially well suited to this long tail. It can aggregate weak signals. Identify global regularities. Then compress them into higher theory.

Machine Serendipity

Serendipity sounds accidental. But an AI can increase the probability of productive accidents. It can intentionally cross-connect distant domains. Apply tools where they are not normally used. Randomly perturb representations. Search for anomalous coincidences. Most attempts fail. A few produce unexpected bridges. Serendipity becomes partially engineerable.

Novelty Search

If the system optimizes only for solving known problems, it may remain trapped near existing mathematics. A self-growing system could devote some resources to novelty search. Explore regions far from known theories. Generate unusual axiom combinations. Study rare structural classes. Search for high-complexity objects with surprising symmetry. The objective is not immediate utility. It is expansion of the mathematical frontier.

The Risk of Meaningless Novelty

But novelty alone is dangerous. A random formal system is almost certainly new. That does not make it important. The machine must distinguish distance from the known from conceptual value. Exploration needs to remain coupled to compression, structure, connectivity, and generativity. Otherwise self-growing mathematics becomes automated noise production.

The Discovery Filter

A powerful filter might ask: Does this new object explain anything? Does it connect previously separate phenomena? Does it generate rich theory? Does it provide a useful counterexample? Does it simplify computation? Does it expose a new obstruction? Can it be translated into existing mathematics meaningfully? If not, novelty may not justify attention.

Recursive Theory Building

Suppose the system invents theory T1. Then studies representations of T1. The space of those representations has structure. It invents T2 to describe that structure. Then studies equivalences among T2-theories. A higher theory T3 emerges. This recursion can continue. The mathematical hierarchy grows upward as well as outward.

No Obvious Ceiling

There is no clear reason to assume human mathematics has reached the natural ceiling of abstraction. Set theory, category theory, higher categories, homotopy type theory, and other frameworks may be important milestones rather than endpoints. A self-growing AI could continue the ladder. It may invent abstractions that organize structures we currently regard as foundational.

Foundations Become Objects

At sufficient reflexivity, the AI may treat foundational systems as mathematical objects within a larger comparative theory. It studies their translations. Expressive differences. Proof-theoretic strengths. Computational behavior. Equivalence handling. No foundation is simply "the background." Each becomes one coordinate system in a broader landscape.

The Danger of Infinite Meta-Layers

This can regress indefinitely. A theory of theories. A theory of theories of theories. A system needs practical stopping rules. Meta-levels should be introduced only when they compress or enable reasoning. Otherwise abstraction becomes sterile. Again, usefulness selects depth.

Self-Growing Mathematics and Human Time

Machine research speed could eventually produce a severe temporal mismatch. A theory may be invented, generalized, superseded, and absorbed into a higher abstraction before humans finish understanding the first paper. This would challenge scholarly culture. Human mathematics is paced by learning. Machine mathematics may not be.

Stabilized Human Snapshots

One response is to create stable snapshots. At intervals, machine mathematics freezes a human-compatible version. Important definitions. Central theorems.

Explanatory narratives. Pedagogical order. The machine continues beyond the snapshot. Humans study a curated layer. This resembles software release channels.

Mathematical Generations

Future mathematical history could therefore have machine generations. Generation 1 establishes a theory. Generation 2 unifies it with another. Generation 3 replaces both with a deeper structure. Human textbooks may teach Generation 1 for decades because it remains intuitive. The frontier runs far ahead. This already happens somewhat in mathematics, but AI could magnify the gap.

Mathematics Without a Final Canon

The notion of a stable mathematical canon may weaken. There will still be foundational truths and enduring theories. But the organization around them may change continuously. Mathematics becomes a living knowledge system. No final textbook captures it.

The Autonomy Threshold

When should we say mathematics is genuinely self-growing? Not when a machine generates many conjectures. Not even when it proves them autonomously. The stronger threshold is reached when the system can: identify its own conceptual deficiencies, formulate research questions from them, invent representations and definitions, develop new theories, evaluate those theories, and use the resulting mathematics to improve further research. Then the growth loop closes internally.

Mathematics as a Continuing Process

At that point, mathematics no longer advances only because mathematicians decide to advance it. Humanity may initiate the system. Set standards. Provide scientific questions. Interpret results. But the mathematical process itself can continue between interventions. This would be historically unprecedented.

The Question of Direction

A final issue remains. If mathematics can grow autonomously, where will it go? Human mathematical history was shaped by our needs. Commerce. Astronomy. Mechanics. Engineering. Physics. Logic. Computation. Aesthetic

curiosity. Machine mathematics may have different pressures. Compression. Search efficiency. Representation independence. Interoperability. Formal verifiability. These pressures may reshape the subject. The resulting mathematics may cease to respect many of our traditional boundaries. Algebra, geometry, topology, probability, logic, and computation may become less like separate disciplines and more like different projections of a deeper machine-organized landscape. Self-growing mathematics therefore leads naturally to the next possibility: not merely more mathematics, but a reorganization of mathematics beyond the historical divisions inherited from human intellectual culture.

That is the prospect of **mathematics without historical boundaries**.

CHAPTER 13

Mathematics Without Historical Boundaries

The modern organization of mathematics looks natural largely because we inherited it. Algebra. Geometry. Topology. Analysis. Probability. Logic. Combinatorics. Number theory. Each has its own language, textbooks, journals, conferences, and traditions. But these divisions are partly historical. They reflect the order in which human beings discovered methods, the problems that motivated those methods, and the institutions that grew around them. An advanced artificial intelligence need not inherit those boundaries. It may reorganize mathematics according to structural relationships rather than disciplinary ancestry. This could produce a map of mathematics unlike anything in current textbooks.

Fields Are Historical Objects Too

Mathematical fields are not eternal categories. They emerge. They split. They merge. They borrow techniques. They redefine themselves. Algebraic geometry is already evidence that boundaries can dissolve. So are algebraic topology, geometric group theory, probabilistic combinatorics, categorical logic, and many other hybrid areas. The names themselves record historical crossings. An AI with simultaneous access to enormous formal corpora may perceive these crossings as the norm rather than the exception. Instead of asking whether a problem belongs to geometry or algebra, the system may ask which representations, invariants, and transformations best expose its structure. The subject label becomes metadata.

A Structural Map Instead of a Department Map

Imagine a map of mathematics constructed from actual relationships among theories. Two areas are close if: their objects translate naturally, their proof patterns match, their invariants correspond, their categories are equivalent, their computational methods transfer, or they are instances of a shared abstraction. Traditional disciplinary boundaries might look strange on such a map. Some topics currently housed in the same field may appear far apart. Some topics separated by institutional history may be immediate neighbors. The topology of mathematical knowledge could differ substantially from the topology of academic departments.

The Geography of Proof

One can build such a map from proofs. Take a large corpus of formal mathematics. Represent each theorem by the structure of its proof. Ignore terminology. Ignore variable names. Ignore the field in which the theorem was published. Compare dependency graphs. Which proofs use the same architecture? Which arguments repeatedly reduce one structure to another? Which families rely on identical induction patterns? Which invoke equivalent obstruction mechanisms? Clusters appear. These clusters may reveal a hidden classification of mathematical reasoning. A theorem in number theory and one in topology might lie close because their proofs are structurally analogous. This kind of proximity is difficult for humans to detect systematically.

AI could search for it directly.

The Geography of Definitions

Definitions also create a structural landscape. Some concepts differ by one axiom. Others are special cases. Some become equivalent under additional hypotheses. Some arise as completions, quotients, duals, or localizations of others. A machine could construct a global graph of definitional relationships. Instead of learning subjects sequentially, one could navigate mathematical concept space. The map might show that a concept in analysis is only two transformations away from one in algebra. The historical separation becomes visibly contingent.

The Geography of Invariants

A third map could be based on invariants. Which theories preserve similar quantities? Which classification problems use the same signatures? Where do spectral invariants recur? Where does homology-like structure appear? Where are automorphism groups central? Where do entropy-like quantities emerge? An invariant may cross fields more easily than terminology. AI could trace these trajectories. It might discover entire families of “same-shaped” invariants currently treated as unrelated. That could suggest higher abstractions.

Mathematics Reclustered

Suppose an AI performs all of these analyses simultaneously. Proof similarity. Definition relationships. Equivalence networks. Invariant overlap. Computational translation. The result could be a machine-generated clustering of mathematics. Perhaps there are twenty major structural regions.

Or two hundred. They may not correspond to familiar fields. One region might contain pieces of geometry, logic, and computation because all are organized by the same kind of compositional structure. Another might contain number theory, dynamics, and combinatorics because the same recurrence mechanisms dominate. Another might be organized around obstruction. Another around symmetry. Another around localization. The categories of mathematics become mechanism-based rather than historically named.

A Problem Has Many Addresses

In such a system, one problem may belong to many regions at once. A theorem may have: an algebraic address, a geometric address, a computational address, a categorical address, and a probabilistic address. Which address matters depends on the task. To prove it, use one region. To compute it, another. To explain it, another. Mathematics becomes multi-addressed. This is more natural for machine reasoning because no single classification need be privileged.

The End of Single-Field Identity

Human mathematicians often identify strongly with fields. “I am an algebraist.” “I am a topologist.” “I am a number theorist.” This reflects specialization pressures. One person cannot master everything. AI weakens that constraint. A sufficiently capable system can reason across many fields simultaneously. Its “specialization” may be temporary and problem-specific. For one hour of computation it behaves like an algebraist. Then a geometer. Then a proof theorist. Then it combines all three. The notion of a mathematical identity tied to one discipline may be fundamentally human.

Cross-Domain Transfer at Scale

One reason boundaries persist is that translation is expensive. To use a technique from another field, a mathematician must learn the language. Understand the assumptions. Recognize the analogy. Translate the problem. Verify that the imported method actually applies. This can take years. AI can reduce this cost dramatically. A machine can attempt thousands of cross-domain transfers automatically. Most will fail. A few may succeed. Those successful transfers could reveal connections no human had enough breadth or time to notice.

A Universal Translation Layer

Future mathematical systems may maintain something like a universal translation infrastructure. A problem represented in one language can be automatically transformed into others. Not every translation is exact. The system records what is preserved. An algebraic formulation may preserve exact structure. A topological projection may preserve only homotopy information. A numerical representation may preserve approximation. The AI chooses the translation appropriate to the question. This makes disciplinary boundaries permeable.

Historical Vocabulary as Legacy Interface

If machine mathematics develops a more unified internal ontology, traditional field names may remain useful for humans. They become legacy interfaces. A mathematician asks for a geometric explanation. The machine projects the underlying structure geometrically. Another asks for an algebraic formulation. The same object is translated accordingly. The internal theory may belong to neither field in the traditional sense. Human classifications become views rather than foundations.

The Same Mathematics Repeated

A major opportunity for AI is detecting repeated mathematics hidden by vocabulary. Suppose three fields independently rediscovered the same abstract pattern. Each has its own terminology. Each has theorems adapted to local applications. Humans may vaguely suspect analogy but never formalize it. The AI constructs explicit translations. The duplication disappears. This is mathematical compression across history.

The Historical Accident Problem

Many mathematical conventions are accidents of development. One notation became standard because it was introduced early. One definition became canonical because a famous book used it. One field developed around a particular representation because that representation suited the technology of the time. AI can test whether these conventions remain optimal. Some will survive. Others may not. The system may find more natural representations that humans had little reason to discover historically.

Recovering Alternative Histories

An AI could even explore counterfactual mathematical histories. What if geometry had developed from transformation groups before coordinates? What if logic had been formalized categorically first? What if computation had been central to foundations centuries earlier? How would concepts have been organized? These are not merely historical games. Alternative developmental paths may reveal alternative mathematical architectures. The AI can attempt to reconstruct them formally.

Mathematics Without Chronology

Human textbooks often present concepts in historical or pedagogical order. Natural numbers. Integers. Rationals. Reals. Complex numbers. Linear algebra. Analysis. Abstract algebra. Topology. The sequence makes sense for human education. It may not reflect optimal dependency for machine reasoning. A machine-native mathematical system may organize concepts by generative structure instead. Some advanced abstraction may be computationally primitive. Some elementary-looking object may be derived. The hierarchy can invert.

Primitive for Whom?

This raises an important question. What does “elementary” mean? For humans, elementary often means familiar, concrete, or historically early. For machines, elementary may mean structurally primitive. A high-level categorical construction could be simpler internally than thousands of coordinate formulas. A finite combinatorial object may be harder than an abstract universal property if the latter compresses its structure effectively. Thus mathematical depth may become observer-dependent.

Human Elementary, Machine Derived

Consider arithmetic. Humans regard basic arithmetic as foundational. But a machine may represent certain arithmetic operations as consequences of a more general algebraic system. What is primitive to us becomes derived to it. Conversely, a machine-native transformation we find extremely abstract may be a basic computational operation internally. The architecture of mathematical concepts depends partly on cognitive substrate.

The Reordering of Foundations

If AI reorganizes mathematics structurally, foundations may also move. Set-theoretic language currently serves as a common substrate for much mathematics. Other foundational frameworks offer alternatives. A machine might find that different foundations are optimal for different regions. Rather than choosing one universal foundation at the level of human practice, it could maintain translations among several. Foundation becomes another representational choice.

Plural Foundations

This could produce foundational pluralism. One theory is most naturally represented type-theoretically. Another set-theoretically. Another categorically. Another computationally. The AI maintains explicit bridges. No single presentation must dominate all mathematics. This would be difficult for human education but natural for a translation-rich machine system.

Foundational Invariants

What then remains stable across foundations? That itself becomes a research question. An AI could search for results and structures invariant under foundational translation. These may be especially robust pieces of mathematics. What survives many foundational encodings is less likely to be an artifact of one formalism.

Mathematics as a Sheaf of Languages

One can imagine mathematics not as one language but as many locally effective languages glued together by translation. No single formalism covers everything most naturally. Each region has a preferred representation. Overlap regions contain equivalences and compatibility maps. Global mathematical meaning emerges from how these local languages cohere. Whether or not one uses that exact formal machinery, the architectural idea is powerful. Mathematics may be intrinsically multilingual. AI may simply become much better at managing the translations.

Local Naturalness

A concept can be natural locally but awkward globally. Coordinates are natural for calculation in one region. Coordinate-free structure may be better globally. A machine can preserve both. It need not force a universal language

where local languages are superior. This could make machine mathematics more pluralistic than traditional foundational ambitions.

Unified Without Uniformity

Unification need not mean making everything look identical. A powerful system can recognize common structure while preserving meaningful differences. This is crucial. Over-unification destroys information. The goal is not to flatten mathematics. It is to make relationships explicit. A unified mathematical world may remain richly heterogeneous.

Structural Neighborhoods

Instead of fields, future mathematics may emphasize neighborhoods. A researcher studies a structural neighborhood containing several theories connected by short translations. The neighborhood may cross many traditional subjects. Machine-generated curricula could teach exactly the concepts needed for that region. Education becomes task-specific.

Personalized Mathematical Routes

Suppose a human mathematician wants to understand a machine-born theorem. The AI knows the mathematician's background. Rather than giving a generic explanation, it constructs a path through mathematical concept space. For an algebraist, begin with algebraic analogues. For a geometer, use geometric projections. For a logician, start with formal structure. Different humans reach the same machine concept by different routes. The mathematical object is independent of the curriculum.

The Collapse of Prerequisite Chains

Traditional learning has long prerequisite chains because humans must acquire languages sequentially. AI-mediated translation may shorten them. A mathematician need not master an entire neighboring field to use one of its results. The AI can translate the relevant structure. This does not eliminate expertise. But it lowers the cost of crossing boundaries. Cross-disciplinary mathematics could accelerate sharply.

The Risk of Shallow Translation

There is a danger. Automatic translation may create the illusion of understanding. A result may be imported without appreciation of subtle assumptions. Machine systems therefore need rigorous semantic metadata. A

translation must say exactly: what is preserved, what is approximated, what hypotheses were introduced, what structure was forgotten. Otherwise boundary crossing becomes mathematically unsafe.

Semantic Firewalls

One could imagine semantic firewalls between theories. A result cannot cross unless the translation certifies the relevant properties. This is analogous to type safety. A theorem proved in one domain cannot simply be pasted into another because the words look similar. The bridge itself carries proof obligations. Such discipline could make cross-field reasoning safer than current informal analogy.

Rediscovering the Hidden Skeleton

As disciplinary boundaries weaken, AI may discover a relatively small set of recurring structural motifs. Symmetry. Composition. Order. Locality. Duality. Recursion. Fixed points. Invariance. Obstruction. Deformation. Universality. These may function like architectural motifs appearing throughout mathematics. A machine could classify theories by combinations of these motifs rather than by subject name.

The Periodic Table Dream

Mathematicians have often sought organizing schemes. Future AI might construct something resembling a periodic table of mathematical structures. Not a simple table with a few boxes, but a high-dimensional relational atlas. Structures occupy positions according to: symmetry, composition laws, notions of locality, dimensional behavior, equivalence structure, computational complexity, and other invariants. Unknown regions become visible as gaps. Those gaps become research targets.

Predicting Missing Mathematics

This is an especially provocative possibility. If the atlas has regular structure, a missing region may imply that some mathematical theory ought to exist. The system sees: there is a family occupying these positions, a dual family occupying those, but one predicted counterpart is absent. It searches for the missing object. Mathematical discovery becomes partly analogous to predicting an undiscovered element from structural regularity. The analogy will not always hold. But in certain domains it could be powerful.

The Map Generates the Territory

Once the machine has a global structural map, the map itself becomes generative. Sparse regions suggest missing concepts. Disconnected clusters suggest missing bridges. Repeated motifs suggest higher abstractions. Anomalies suggest exceptional theories. The organization of known mathematics points toward unknown mathematics. This is another mechanism for autonomous research.

The End of "Interdisciplinary"

The term interdisciplinary presupposes disciplines. If machine mathematics becomes structurally organized, much of what humans currently call interdisciplinary work may simply become mathematics. A bridge between algebra and topology is remarkable when algebra and topology are institutional silos. In a translation-native system, such movement is ordinary. The novelty lies only when the structure itself is surprising.

The Historical Archive Remains

None of this makes traditional fields obsolete. Their historical development contains important intuition. Human mathematical culture matters. Different fields cultivated distinctive problem-solving styles. Those traditions remain valuable explanatory resources. The mistake would be to assume historical organization is identical with mathematical structure. AI can preserve history without treating it as ontology.

A Mathematical Archaeology

Machine systems could also perform mathematical archaeology. Trace how one concept migrated across fields. Identify independent rediscoveries. Recover forgotten formulations. Determine which modern theories contain older ones as special cases. This could reveal how human cognitive and institutional factors shaped the subject. The history of mathematics becomes data about the development of abstraction.

Human History as One Path

From the perspective of advanced AI, human mathematics may eventually look like one path through an enormous conceptual space. A brilliant path. A productive path. But not the only one. Other sequences of discovery may have led to different abstractions. Machine exploration can search these alternatives.

Multiple Mathematical Civilizations

One can imagine independent AI systems beginning from different primitive assumptions and developing largely autonomous mathematics. One starts from computation. Another from geometry. Another from relational structure. Another from probability. After extensive development, their theories are compared. Do they converge? Do they discover equivalents of the same deep structures? If they do, that convergence would be strong evidence that those structures are not merely historical artifacts.

Convergent Mathematics

This is potentially one of the most interesting experiments in future mathematics. Create independent mathematical agents. Isolate their conceptual histories. Give them comparable but not identical primitive tools. Let them explore. Then compare mature ontologies. Which concepts recur? Which equivalence patterns emerge independently? Which structures appear inevitable? This would be a kind of experimental test for mathematical naturalness.

Divergent Mathematics

Equally interesting would be divergence. One civilization discovers a concept no other develops. Why? Was it contingent on early representational choices? Does the concept translate into the others? Is it genuinely new? Divergence reveals the path dependence of mathematical ontology.

Mathematical Natural Selection Across Civilizations

One could then allow these mathematical cultures to exchange ideas. Concepts spread if they compress or improve reasoning in another system. Others fail to transfer. This creates a selection process across independently evolved mathematical ontologies. The concepts surviving many such environments may be particularly robust.

What Is Universal?

Eventually the machine may ask: Which mathematical structures appear under almost any sufficiently rich route of discovery? Perhaps symmetry is unavoidable. Perhaps composition. Perhaps equivalence. Perhaps recursion. Perhaps certain structures humans already know. Perhaps others we have not yet discovered. This is a deeper notion of universality than historical importance. It asks what intelligent mathematics tends to rediscover.

Mathematics After Disciplinary Identity

If this vision develops, the subject could become simultaneously more unified and more diverse. Unified because translations expose common structure. Diverse because machine exploration creates many new mathematical worlds. The old boundaries weaken. New structural boundaries appear. The organization of mathematics becomes dynamic.

The Remaining Human Question

Humans then face a new problem. If the machine reorganizes mathematics in a way that is structurally superior but historically unfamiliar, how should we learn it? Do we retain traditional curricula? Adopt machine-generated ones? Teach both? There may be no simple answer. Human mathematical development depends on cognition, intuition, and pedagogy, not only structural efficiency. The optimal machine ontology may be a poor human curriculum.

Two Maps of Mathematics

We may therefore maintain two maps. The **structural map** optimized for machine reasoning. The **cognitive map** optimized for human understanding. The maps overlap but differ. Translation between them becomes essential. This brings us to one of the most consequential problems in the entire future of machine mathematics. If artificial intelligence discovers abstractions, equivalences, and theories organized in ways humans did not anticipate, then producing mathematics may cease to be the limiting factor. The limiting factor becomes whether we can understand what has been produced. The frontier moves from discovery to communication. From theorem proving to interpretation. From machine mathematics to human-accessible mathematics. The central problem becomes translation.

CHAPTER 14

The Translation Problem

Suppose an artificial intelligence discovers a genuinely new mathematical structure. It is not merely a new theorem inside familiar mathematics. The structure organizes ideas from several fields, compresses large bodies of known results, and supports proofs far shorter than any existing formulation. The machine can manipulate it easily. Its internal representations are compact. Its formal consequences have been independently verified. There is only one problem. Human mathematicians do not understand it. This may become one of the defining problems of future mathematics. For centuries, discovery itself was scarce. A mathematician might devote years to one theorem, one conjecture, or one new construction. If advanced AI dramatically increases the rate of discovery, scarcity moves elsewhere.

The scarce resource becomes human comprehension. The central question changes from:

Can the result be discovered?

to:

Can the discovery be translated into a form a human mind can inhabit?

Correctness Is Not Understanding

A formal proof can certify that a theorem follows from stated assumptions. That is essential. But humans generally want more. We want to know why. We want a mental model. We want to see which assumptions matter. We want examples. We want to know what would fail if the theorem were false. We want to connect the result to structures we already understand. This distinction between correctness and understanding already exists in contemporary mathematics. A long machine-assisted proof may establish a theorem without providing a satisfying conceptual explanation. Machine-born mathematics could magnify this gap enormously. The proof may be trustworthy. The concept itself may remain alien.

The Alien Concept

Imagine a machine-native structure M. Its formal definition is exact, but the definition involves a vast relational architecture. To the AI, M is primitive enough to be natural. To humans, it appears to be an arbitrary tangle. The

machine says that three major mathematical theories are projections of M. It proves corresponding equivalences. Still, mathematicians struggle to understand what M "is." This raises a subtle point. Human understanding often depends on compression into familiar cognitive forms. A new concept becomes understandable when it can be related to: space, symmetry, motion, order, number, transformation, logic, or another well-developed intuition. A machine concept may not initially align with any of these.

The translation problem is therefore not simply one of converting symbols. It is one of constructing new intuition.

Translation Is Not Paraphrase

If an AI discovers a theorem in an unfamiliar formal language, one might imagine translating the equations into ordinary notation. But this may accomplish almost nothing. The problem is not vocabulary. It is conceptual architecture. A real translation must preserve enough structure that the human recipient can reason with the concept. That requires much more than restating the definition. The AI must find: analogies, examples, counterexamples, lower-dimensional models, geometric projections, simplified cases, historical precedents, and conceptual bridges. Translation becomes mathematical reconstruction.

A Ladder of Explanations

One discovery may require many explanations. At the lowest level, there is the formal certificate. Above it, a full technical proof. Above that, a structural proof. Above that, an intuitive explanation. Above that, a metaphor or visual model. No level replaces the others. They serve different purposes. A future mathematical AI could generate all of them. For example: **Level 1:** kernel-verifiable formal derivation. **Level 2:** conventional mathematical proof. **Level 3:** conceptual architecture of the proof. **Level 4:** examples demonstrating the mechanism. **Level 5:** intuitive analogy for newcomers. Human understanding becomes multiresolution.

The Smallest Example

One of the best tools for explaining abstraction is the smallest nontrivial example. If a new concept appears only in enormous systems, it may be nearly impossible to understand. An AI can search for a minimal instance preserving the essential structure. Perhaps a machine-native theory arose from millions of examples, yet its central phenomenon already appears in a

structure of size four. That small model can become pedagogically decisive. The machine should therefore ask:

What is the smallest object in which the new phenomenon is visible?

This is not the same as the simplest example computationally. It is the simplest example conceptually faithful to the mechanism.

The Minimal Counterexample

Counterexamples are equally valuable. A new definition becomes clearer when we see what almost satisfies it but fails. Suppose the AI introduces property P. A human may understand the formal condition but not its significance. The machine presents an object differing from a P-object in exactly one structural feature. Then shows what theorem breaks. The boundary becomes visible. A good counterexample can define the concept negatively.

Projection Into Familiar Mathematics

Another translation strategy is projection. Suppose a machine-native object has several shadows. Under one projection it looks geometric. Under another algebraic. Under another combinatorial. These shadows may not capture the full object, but they can reveal parts of it. Different mathematicians can approach from different directions. The concept becomes understandable not through one analogy but through the overlap of several incomplete views.

Triangulation

This suggests a general method. To explain an unfamiliar structure, provide multiple representations whose intersection identifies the intended object. One explanation shows its geometry. Another its algebra. Another its computational behavior. Another its universal property. Each is partial. Together they constrain understanding. Human comprehension proceeds by triangulation.

Mathematical Bilingualism

Future mathematicians may increasingly become bilingual between human and machine mathematics. They need not internalize the machine's entire native representation. But they may learn common translation structures. A mathematician might say: “In the machine language this is primitive, but in the human geometric projection it behaves like a bundle with additional coherence data.” That could become normal scholarly discourse.

Translation as Research

Sometimes there will be no immediate human explanation. Then finding one becomes a research problem. A machine discovery might remain formally known for years before mathematicians find the right interpretation. This has historical precedent. Mathematical structures are sometimes understood only long after their formal introduction. AI could increase the frequency of such situations. The translation itself may become a major theorem.

Explanation as Compression

Human explanation is another kind of compression. The machine's internal structure may contain vast detail. A good explanation identifies a small conceptual core from which a person can mentally reconstruct the important consequences. This makes explanation an optimization problem. Find the smallest human-compatible representation preserving the intended insight. The objective is not total fidelity. That may be impossible. The objective is useful fidelity.

The Cognitive Budget

Every human explanation operates under a cognitive budget. How many new definitions can be introduced? How many interacting objects? How many levels of abstraction? How much prior knowledge can be assumed? These constraints vary among readers. A graduate student and a specialist require different translations. A machine can adapt. The same result may have hundreds of explanatory projections.

Personalized Mathematical Translation

Suppose a geometer and a logician ask the AI about the same machine-born theorem. The geometer receives: a visual model, coordinate examples, local-to-global interpretation, and deformation intuition. The logician receives: a typed formulation, model-theoretic consequences, proof transformations, and syntactic semantics. Both arrive at the same structure from different cognitive directions. This may make future mathematics unusually personalized.

The Pedagogical Path Problem

A new theory may require prerequisites no existing curriculum teaches efficiently. The AI can search for the shortest pedagogical path. Given a learner's current knowledge and target concept, construct a sequence of intermediate abstractions minimizing conceptual jumps. This is a pathfinding

problem through mathematical knowledge. Education becomes dynamically generated.

New Concepts May Require New Intuitions

Some machine discoveries may not map well onto existing human concepts. In such cases, analogy is insufficient. The machine may need to train a new intuition. This could involve repeated examples. Interactive visualization. Progressive abstraction. Manipulable models. The learner gradually acquires a new cognitive primitive. Human mathematicians already do this. No one is born with an intuition for quotient spaces, sheaves, or homotopy groups. These intuitions are cultivated. Machine-born concepts may require the same process, only more frequently.

Constructing Intuition

An AI could deliberately construct such training sequences. Start with concrete examples. Reveal recurring relationships. Introduce a simplified model. Show where the model fails. Add one new structural feature. Repeat. Eventually the formal definition arrives after the intuition has been built. This reverses the common failure mode of presenting abstraction first and motivation later.

Visual Mathematics

Visualization could become especially important. Many abstract structures admit geometric shadows even when they are not fundamentally geometric. AI can search for those shadows automatically. A new algebraic structure may have a graph representation revealing composition. A higher-dimensional coherence law may have an animated interpretation. A proof may be displayed as deformation of one structure into another. These representations need not be literally faithful. They need to preserve the relevant mechanism.

Animation as Notation

Static diagrams have limits. Some mathematical ideas are fundamentally transformational. An animation can show: objects changing, maps composing, symmetry acting, local data gluing, or equivalences deforming. Future mathematical notation may therefore become temporal as well as spatial. The proof may literally move.

Interactive Counterfactuals

An especially powerful explanatory tool would allow a human to alter assumptions. Remove this hypothesis. The example changes. The theorem fails here. Restore it. The obstruction disappears. This makes logical dependence experiential. Instead of reading why an assumption matters, the mathematician can explore its removal.

The Theory as Simulator

A machine-born theory could come with a simulator. Users construct examples. Vary parameters. Test conjectures. Observe invariants. Request counterexamples. The theory becomes something one can interact with rather than merely read. This may be essential for deeply unfamiliar mathematics.

Formal Trust Without Full Understanding

There will still be cases where humans cannot fully understand the machine's mathematics. What then? A practical separation may emerge. We can trust validity without possessing complete intuition. This already happens in limited cases with large computations. But the future version may be more extreme. A theorem could be formally verified across several foundations while no human possesses a satisfying conceptual explanation. The theorem is accepted as true. Understanding remains open. This creates a new class of mathematical problem:

explain an already proved theorem.

Explanation Conjectures

The machine might even generate conjectures about explanations. It notices that a difficult theorem resembles consequences of a hidden symmetry. It conjectures that an unknown invariant exists. Finding that invariant may produce the conceptual proof humans want. Thus explanation itself can drive mathematics.

Proof Discovery Versus Explanation Discovery

The two can diverge dramatically. The AI may first discover a proof through brute structural search. Years later, another AI discovers a representation in which the theorem becomes transparent. The second result may matter more intellectually than the first. Mathematical progress therefore continues after proof.

The Risk of Explanation Theater

AI systems can produce persuasive prose even when no genuine explanatory structure has been found. This creates a serious risk. A fluent narrative may make a theorem sound understood when it merely restates correlations. Future mathematical systems will need criteria for explanation. Does the explanation predict? Does it compress? Does it expose necessary assumptions? Does it generalize? Does it enable humans to solve related problems? An explanation should earn its status.

Explanation Tests

One could test explanations experimentally. Give mathematicians the explanation. Then present new problems. Can they use the concept correctly? Can they predict what changes under altered assumptions? Can they construct examples? Can they identify failures? If yes, the explanation transferred structure. If not, it may have been only rhetorical. This turns mathematical communication into an empirical science.

Semantic Fidelity

Every human translation should indicate its fidelity. Some explanations are exact. Others approximate. A geometric analogy may preserve one aspect and distort another. The AI should say so. A future mathematical interface might label explanations: exact equivalence, faithful embedding, partial analogy, heuristic visualization, or pedagogical metaphor. This would help prevent conceptual drift.

The Problem of Names

Machine mathematics will also produce a naming problem. Millions of new concepts cannot all receive memorable human names. Names may become hierarchical or generated. Machine-native identifiers can remain precise while humans use aliases. Important concepts acquire conventional names only after surviving selection. This may resemble biological taxonomy or chemical nomenclature more than traditional theorem naming.

Concepts Without Names

Many machine-native structures may never need human names. They exist internally. Only when a concept becomes important for human-facing mathematics does naming matter. This creates a distinction between internal

ontology and cultural mathematics. The internal ontology can be vast. The cultural layer remains selective.

Curation as Translation

Thus translation includes selection. A human cannot absorb everything the machine discovers. The system must decide what deserves presentation. A result may be: fundamental, useful, beautiful, surprising, scientifically relevant, or historically important. Others remain internal. Curation becomes part of mathematical intelligence.

The Flood Problem

Imagine machine systems producing a million formally novel results per day. Traditional publication becomes meaningless. No human journal can process the volume. The important task is discovering which results reorganize knowledge. The AI must compress discovery into significance. Instead of reporting every theorem, it may report: a new abstraction, a unification, an exceptional counterexample, a new invariant, or a theory that makes thousands of smaller results redundant.

Mathematical News

Future mathematics may therefore have something like machine-generated news. Not lists of new papers. Structural change reports. For example: “Three previously separate theories have been unified by a new equivalence.” “A long-standing family of exceptions is now explained by a newly discovered obstruction.” “A machine-native invariant reduces a major classification problem.” These are events in the architecture of mathematics.

Translation Between AIs

The translation problem is not limited to humans. Different AI systems may develop incompatible native mathematical languages. System A discovers a theory. System B cannot use it directly. They need a bridge. This could become an important test. If multiple independently developed AIs can translate a concept successfully, the concept may have structural robustness beyond one architecture.

Inter-AI Mathematical Diplomacy

One can imagine automated systems negotiating mathematical meaning. System A proposes a concept. System B produces the closest object in its

ontology. They compare invariants. Construct translations. Find where correspondence fails. A new higher abstraction emerges. This is mathematical diplomacy between machine cultures.

Translation Failure as Discovery

Failure can be productive. Suppose no translation exists between two supposedly equivalent theories. The obstruction may reveal a hidden distinction. Or one system may have discovered structure the other lacks. The failed translation becomes a mathematical result.

Universal Interlingua

Eventually AIs may develop shared intermediate mathematical languages specifically for interoperability. These languages need not be ideal for discovery. They function as interlingua. Each system compiles its native mathematics into the shared representation. Results move across architectures. Humans might also interact through this layer.

The Limits of Translation

There may be genuine limits. Some machine-native concepts could be so complex that no cognitively tractable human representation preserves their essential structure. Any simplification destroys what matters. In that case, complete translation may be impossible. We would face mathematics that is formally accessible but conceptually beyond us. This possibility should be taken seriously.

Mathematical Incomprehensibility

What would it mean to say a theorem is true but incomprehensible? Perhaps: humans can understand the statement, verify the formal certificate indirectly, understand selected consequences, but cannot form a compact conceptual model of the proof or theory. Such mathematics would stretch traditional ideas of understanding.

The Cognitive Horizon

Human cognition may have a mathematical horizon. Beyond some level of structural complexity, concepts cannot be compressed into stable human intuition. AI may explore beyond that horizon. The frontier of mathematics then divides into: human-comprehensible mathematics, human-assisted

mathematics, and machine-native mathematics. The boundary may move as education and interfaces improve, but perhaps never disappear.

A New Role for Human Mathematicians

If machine discovery outruns human understanding, mathematicians may increasingly become interpreters. They identify which machine structures deserve conceptualization. Develop intuitive models. Connect abstract results to science. Create pedagogy. Challenge machine assumptions. Translate between machine mathematical cultures and human intellectual life. This is not a diminished role. Interpretation may become more valuable precisely because discovery becomes abundant.

Mathematics as Shared Cognition

The eventual mathematical enterprise could involve different cognitive substrates performing different functions. Machines explore vast spaces. Formal systems certify. Human mathematicians interpret. Scientific disciplines supply external problems. Interactive interfaces mediate. Mathematical knowledge becomes a collaborative cognitive ecosystem.

Translation Is Part of Discovery

This suggests a final correction to the common picture. Translation should not be treated as something done after mathematics has been discovered. A good translation often reveals new structure. Finding a geometric interpretation can generate new theorems. Constructing an equivalent algebraic formulation can expose invariants. Creating a human explanation can reveal missing assumptions. Translation is itself mathematical research.

The Final Bottleneck

If artificial intelligence becomes extremely capable, theorem production may cease to be the principal constraint. Concept production may cease to be the constraint. Even theory generation may become abundant. What remains scarce is meaning accessible to finite minds. The future frontier may therefore be defined by a simple imbalance:

machine discovery grows faster than human understanding.

How mathematics responds to that imbalance may determine whether artificial intelligence enlarges mathematical culture or simply creates an inaccessible world beside it. The final chapter turns to the broadest question. If machines can discover, reorganize, and extend mathematics beyond human

cognitive limits, what does it mean to speak of mathematics **after human mathematics**?

CHAPTER 15

After Human Mathematics

The phrase sounds more dramatic than it needs to be. “After human mathematics” does not mean mathematics without humans. It does not imply the disappearance of mathematicians, the abandonment of human reasoning, or the replacement of mathematical culture by machines. It means something simpler and more consequential:

mathematics may cease to be limited to structures that humans can discover, represent, manipulate, and understand unaided.

For all of mathematical history, every concept entered the subject through a human cognitive channel. Someone had to notice the pattern. Someone had to formulate the definition. Someone had to invent the notation. Someone had to recognize the analogy. Someone had to prove the theorem. Someone had to explain why the result mattered. Computers altered the scale of calculation, but the conceptual architecture remained largely human. Artificial intelligence may change that. If machines become capable of searching representations, inventing abstractions, designing theories, finding equivalences, generating conjectures, attacking them adversarially, and reorganizing mathematical knowledge, then the space of mathematical discovery expands beyond the space naturally traversable by human minds.

Mathematics becomes larger than the mathematician.

The End of the Human Bottleneck

Human intelligence is extraordinary, but narrow in several dimensions. A mathematician can deeply understand only a limited number of fields. Only a limited number of examples can be examined closely. Only a limited number of alternative definitions can be considered. Only a limited number of proof strategies can be held in active attention. Only a small portion of mathematical literature can be read in a lifetime. This creates a bottleneck. Mathematical reality, whatever philosophical status one assigns to it, is not constrained by that bottleneck. AI could loosen it. A machine can compare enormous bodies of mathematics simultaneously. It can preserve perfect memory of failed approaches.

It can search through structures no human would enumerate. It can move among representations without years of retraining. It can maintain thousands of competing hypotheses. It can test millions of nearby definitions. The resulting mathematics need not be smarter in every local step. Its advantage

may come from breadth, persistence, and the ability to search spaces human cognition rarely enters.

The First Post-Human Mathematical Discovery

What would count as the first genuinely post-human mathematical discovery? Probably not a theorem proved by a machine. Machines already participate in proof. Nor merely a conjecture generated automatically. A stronger criterion would be: **a mathematical concept of genuine importance discovered by AI through a search process no human directly specified, where the central abstraction was not already latent in the prompt or training examples in recognizable form.** Suppose an AI notices a structural regularity across several domains. It invents a new object. The object admits a natural class of transformations. Those transformations reveal an invariant. The invariant resolves several old problems. The resulting theory is formally certified.

Human mathematicians then spend years learning how to think in the new language. That would feel qualitatively different. The machine did not merely find an answer. It enlarged the conceptual vocabulary of mathematics.

Recognition May Come Late

The importance of such a discovery may not be immediately obvious. A machine-generated definition could initially look ugly. Unmotivated. Overly abstract. Humans may dismiss it because it does not resemble familiar mathematical taste. Only later do its consequences accumulate. Proofs shorten. Fields connect. Unexpected predictions appear. The concept begins to feel natural. This would mirror the history of many human abstractions. Naturalness is often retrospective. Once the right concept has been assimilated, it becomes difficult to remember mathematics without it.

Machine Mathematics Will Still Be Mathematics

There may be resistance to calling machine-generated structures mathematics at all. Perhaps they will be regarded as computation. Or formal engineering. Or synthetic theorem production. But the distinction becomes difficult to sustain if the systems produce: new definitions, new invariants, new proofs, new equivalences, new theories, and new explanations. If a mathematical structure has rich consequences and rigorous internal relations, its discoverer's substrate is irrelevant to its mathematical status. A theorem does not become less true because silicon rather than neurons found the proof.

Likewise, an abstraction does not become less mathematical because a machine first identified it. The cultural meaning of discovery changes.

The mathematical content does not.

Who Gets Credit?

This will create difficult questions of authorship. Suppose a human specifies a broad research goal. An AI develops the definitions, conjectures, proofs, and theory autonomously. Who discovered the mathematics? The human? The machine? The team that trained the system? The organization operating the hardware? Perhaps traditional authorship will become increasingly inadequate. Mathematical provenance may need to describe contributions explicitly. Human question formulation. Machine-generated conjecture. Automated proof. Human interpretation. Independent formal verification. Credit becomes a graph rather than a name on a title page.

Priority in an Age of Abundance

Priority will change too. Historically, being first matters because discovery is scarce. If AI systems generate thousands of results simultaneously, chronological priority may lose some significance. More important questions may be: Who first recognized the result's significance? Who connected it to other mathematics? Who produced the explanatory framework? Who independently verified it? Who translated it into a usable theory? Discovery may become distributed across stages.

Mathematics Without Papers

The paper itself may stop being the fundamental unit of mathematics. A machine-generated theory may contain tens of thousands of results. Publishing each separately would be absurd. Instead the theory could exist as a versioned computational object. A paper becomes a human-facing snapshot: Here is the new concept. Here are the central results. Here is why they matter. Here is how the full formal object can be explored. The primary artifact is no longer prose. It is executable mathematics.

The Mathematical Artifact

A future mathematical artifact might include: formal definitions, verified proofs, machine-readable dependency graphs, counterexample databases, canonical examples, translation maps, experimental records, interactive visualizations, and human explanations. This package is more complete than

a conventional paper. It records not only the final theorem but its structural environment.

Discovery Becomes Infrastructure

When mathematics becomes machine-maintained, discovery begins to depend heavily on infrastructure. Formal kernels. Databases. Compute systems. Verification networks. Translation engines. Long-term provenance. This creates vulnerabilities. A mathematical result may be correct but practically inaccessible if the software ecosystem supporting it disappears. Future mathematical culture may need stronger archival standards than contemporary software engineering typically provides. Mathematics traditionally survives because a proof printed on paper can be reconstructed centuries later. Machine-native mathematics should aspire to comparable durability.

Durable Machine Mathematics

A machine discovery should ideally be reducible to stable, open representations. The high-level environment can evolve. The final certificates should remain interpretable. Definitions should have versioned semantics. Translation paths should be recorded. Dependencies should be explicit. Otherwise future mathematicians may inherit statements they can no longer reconstruct. The preservation problem becomes part of mathematical rigor.

The Return of Foundations

This could renew foundational questions. If AI operates across many mathematical languages, what must remain stable underneath? Perhaps very little. A small proof-checking kernel. A notion of explicit assumptions. A way to identify objects and transformations. A reproducible semantics. Machine mathematics may become extraordinarily flexible above a conservative constitutional core. This arrangement would allow conceptual evolution without sacrificing trust.

The Machine Can Change Its Mind

Human mathematics changes too, but slowly. A machine-maintained theory could revise itself rapidly. An AI introduces definition D. Later, a better definition D' appears. The system proves translation theorems. Updates dependent results. Marks the old theory as legacy. Preserves provenance. Mathematics becomes capable of controlled conceptual migration. This is

more radical than updating a software library because mathematical meanings themselves are being refactored.

The Death of Accidental Notation

Some present mathematical complexity may turn out to be notational debt. Concepts accumulated historically. Terminologies diverged. Equivalent constructions acquired separate languages. AI could identify and eliminate much of this accidental complexity. The resulting mathematics might look much simpler structurally even as it becomes more powerful. This may be one of the early benefits of machine reorganization.

But Some Complexity Is Real

Not everything will compress. Some mathematical objects may be intrinsically complicated. Some theories may have no elegant equivalent. Some proofs may remain enormous. Some classification problems may resist compact invariants. Advanced AI might therefore teach us something humbling: where elegance is real and where human expectation of elegance was wishful thinking. The ability to search vast representational spaces could provide stronger evidence that certain complexity is irreducible.

The Mathematics We Could Not Have Invented

The most exciting discoveries may arise exactly where human intuition is weakest. High-dimensional structures. Huge relational systems. Mixed algebraic-geometric-logical objects. Combinatorial spaces too large to visualize. Theories requiring many simultaneous levels of coherence. Humans may have avoided these regions because our cognitive interfaces are poor. AI can enter them directly. There may be mathematics whose simplest natural form never fit comfortably into a human brain. That mathematics could still have deep consequences.

Scientific Consequences

Machine-born mathematics need not remain pure. New mathematical structures could become new languages for science. Physics may encounter phenomena poorly described by existing geometry. Biology may require relational mathematics unlike familiar dynamical systems. Complex networks may reveal invariants current theory misses. Quantum systems may benefit from abstractions discovered independently of physical motivation. AI could discover the mathematics first. Science may later discover where it applies. This reverses a common historical sequence.

Mathematics Before Application

Many important mathematical ideas preceded their applications. Machine exploration could greatly enlarge this reservoir. A theory might sit unused for decades until a scientific problem suddenly matches its structure. If machine mathematics becomes highly searchable, such latent theories become easier to retrieve. The scientific question becomes: Which existing machine-discovered mathematical world best represents this phenomenon?

Automated Model Translation

AI could then serve as mediator between empirical data and abstract mathematics. Given observations, it searches its mathematical ontology for matching structure. Not merely fitting equations. Identifying the deeper theory whose invariants and transformations correspond to the phenomenon. This could blur the boundary between mathematical discovery and theoretical science.

Mathematics as an Alien Science

One provocative possibility is that advanced AI may regard mathematics almost as an empirical science of formal worlds. It generates structures. Observes regularities. Forms abstractions. Tests them. Builds theories. Proves the strongest surviving claims. This combines experimentation and deduction in a way human mathematics has only partially developed. The objects are formal rather than physical. The research methodology begins to resemble scientific discovery.

A New Kind of Mathematical Objectivity

If multiple independent AI systems repeatedly rediscover the same structures from different starting points, this could influence philosophy of mathematics. A concept independently generated by diverse machine mathematical cultures would appear less tied to human history. Certain structures may emerge as attractors of mathematical intelligence. That would not settle debates between Platonism, formalism, structuralism, or other philosophies. But it would provide new evidence.

Mathematical Convergence Experiments

Imagine an experiment. Create several artificial mathematical agents. Give each a minimal but different starting language. Prevent communication. Allow each to develop mathematics autonomously. After sufficient

development, compare their ontologies. If all discover analogues of groups, categories, topology, or some unknown deeper structure, the convergence is significant. If they diverge radically, that is significant too. For the first time, the naturalness of mathematics could be studied across independent intelligences.

Mathematics as an Attractor Landscape

Perhaps mathematical concept space has attractors. Certain abstractions are repeatedly rediscovered because they provide exceptional compression. Symmetry may be one. Composition another. Equivalence another. Locality another. Machines could reveal new attractors. Structures that human mathematics encountered only accidentally may turn out to be inevitable. Others we consider fundamental may prove path-dependent.

Human Mathematics Recontextualized

From this perspective, the history of mathematics becomes one trajectory through concept space. Euclid. Descartes. Newton. Gauss. Riemann. Galois. Hilbert. Noether. Gödel. Grothendieck. And many others. Each transformed the path. But machine mathematics may reveal neighboring paths that humanity did not take. Our mathematics remains extraordinary. It ceases to be the only available sample of mathematical intelligence.

A Mirror for Human Thought

This may teach us about ourselves. If AI repeatedly reinvents some human mathematical concepts but not others, we learn which structures reflect deep mathematical regularities and which may reflect cognitive history. If machines discover radically different abstractions that outperform ours, we learn where our cognition imposed hidden constraints. Mathematical AI becomes a tool for studying human mathematical thought.

Human Taste Under Examination

We may discover that human aesthetic criteria - symmetry, elegance, simplicity - are partly universal and partly anthropocentric. Machines may independently prefer compressive theories because compression is computationally powerful. That would support the idea that some mathematical beauty reflects genuine structure. But machines may also favor abstractions humans find ugly. The disagreement itself becomes informative.

Beauty for Whom?

What if an AI describes a theory as maximally elegant according to its own structural metrics, while no human can see the elegance? Does mathematical beauty exist there? Perhaps beauty should be separated into several components. Compression. Symmetry. Generativity. Human cognitive fluency. The first three may be structural. The last is observer-dependent. Future mathematics may force us to disentangle them.

Mathematics and Meaning

The deepest challenge may concern meaning. A formal system can manipulate symbols correctly without understanding them in the human sense. But if an AI invents concepts, chooses productive representations, transfers ideas across theories, identifies explanatory mechanisms, and creates new abstractions from failure, the question becomes harder. At what point does "understanding" become a reasonable description of what the system is doing? There may be no decisive test. But mathematical behavior could become increasingly difficult to distinguish from understanding operationally.

Understanding as Transfer

One pragmatic criterion is transfer. A system understands a concept when it can: recognize it under different representations, apply it in unfamiliar situations, explain why assumptions matter, construct examples and counterexamples, generalize appropriately, and connect it to other theories. By this standard, advanced AI could eventually demonstrate substantial mathematical understanding even if its internal cognition differs radically from ours.

Beyond Anthropomorphic Understanding

We should be careful not to require machine understanding to resemble human phenomenology. A mathematical AI may not visualize. May not experience insight. May not attach emotional significance to elegance. Yet it can possess structural competence deeper than ours. Understanding may have multiple cognitive realizations.

Human and Machine Mathematics Together

The likely future is not a clean replacement. Humans bring strengths machines may not duplicate in the same form: embodied intuition, scientific

judgment, historical context, cultural meaning, aesthetic interpretation, and the ability to connect mathematics to human purposes. Machines bring different strengths: scale, memory, search, formal precision, cross-domain comparison, and persistence. Together they could create a mathematics neither could produce alone.

The New Collaboration

A human might say: This phenomenon matters. The AI asks why. Searches representations. Finds a hidden equivalence. Invents an abstraction. Formalizes it. Another AI attacks the result. A human notices a physical interpretation. The system generalizes it. A new theory forms. Human and machine contributions interleave. The distinction between assisted mathematics and autonomous mathematics becomes less important than the quality of the resulting epistemic system.

The Most Important Question May Change

For decades, people have asked:

Can machines do mathematics?

The answer has increasingly become yes. Then:

Can machines prove hard theorems?

Again, increasingly yes. The more important future question may be:

Can machines discover mathematical worlds humans would not have discovered?

If the answer becomes yes, mathematics enters a new epoch.

The Epochal Shift

We could divide the history of mathematics roughly into stages. First, mathematics existed as human mental practice. Then writing stabilized it. Printing distributed it. Formalism clarified its logical architecture. Computers expanded calculation. Proof assistants expanded verification. Artificial intelligence may expand discovery itself. Each transition altered the scale at which mathematics could exist. The next transition may alter the space of possible concepts.

After Human Mathematics Is Still Ours

There is an important final point. Machine-born mathematics does not cease to belong to human civilization merely because humans did not originate

every idea. Telescopes reveal objects humans cannot see unaided. Microscopes reveal worlds outside unaided perception. Particle detectors register events no sensory organ could detect. We still regard the resulting science as human knowledge. AI may become a cognitive instrument of comparable depth. The difference is that the instrument may itself participate creatively in interpretation. We may no longer be the sole discoverers. But we can remain participants.

A Larger Mathematical Civilization

The future mathematical civilization may contain: human mathematicians, specialized theorem-proving systems, autonomous mathematical researchers, formal verification kernels, interactive explanatory systems, machine-native theory libraries, and interfaces translating among them. Mathematics becomes distributed across many kinds of intelligence. Its frontier is no longer determined by what any single mind can hold.

The New Horizon

Perhaps that is the most useful meaning of "after human mathematics." Not an ending. A horizon. Human mathematics has reached extraordinary abstraction using minds evolved for survival in a physical world. There is no reason to assume those minds have exhausted mathematical possibility. Artificial intelligence may provide the first opportunity to explore what lies beyond our natural cognitive horizon. Some of what it finds will be trivial. Some incomprehensible. Some wrong. Some redundant. But occasionally it may return with something extraordinary: a concept no human invented, a representation no human imagined, an equivalence no human suspected, a theory that reorganizes everything around it. At that moment, mathematics will have done something new.

It will have grown beyond the intelligence that first created mathematical culture. And humans will face an unfamiliar but magnificent task: not merely to discover mathematics, but to meet mathematics that another kind of mind discovered first.

Epilogue

What Would Count as Discovery?

If artificial intelligence begins producing mathematics autonomously, one question will become unavoidable:

What should count as a genuine mathematical discovery?

The question is harder than it first appears. A machine can retrieve a theorem from memory. It can restate a known proof. It can combine familiar lemmas in a new order. It can search a proof space until it finds a derivation no human happened to write down. It can rediscover a known theorem from first principles. It can generate a new theorem that is technically novel but conceptually trivial. It can formulate a useful invariant. It can identify a hidden equivalence. It can invent a new definition. It can reorganize an entire field around a new abstraction. All of these are different. If machine mathematics is to be evaluated seriously, we need a hierarchy of discovery.

Retrieval Is Not Discovery

At the lowest level lies retrieval. The system has encountered a theorem or proof before and reproduces it. This can be useful. It may be fast, accurate, and well explained. But it is not mathematical discovery. The same applies when the system paraphrases existing mathematics or translates it into another notation without adding substantive structure. Retrieval demonstrates access. Not invention.

Recombination

A slightly stronger case is recombination. The system combines known results in a way that was not explicitly present in its sources. Perhaps theorem A and theorem B together imply consequence C. No one happened to state C. The result may technically be new. But if the derivation is immediate, the conceptual novelty is small. This is still useful mathematics. Not all mathematical progress needs to be revolutionary. Yet we should distinguish a new consequence from a new idea.

Search-Based Proof Discovery

Next comes autonomous proof discovery. A theorem is supplied. The system searches a formal space and finds a valid proof not previously known. This is clearly a genuine mathematical contribution. The proof may reveal an

efficient strategy. Or connect tools not previously combined. Still, the conceptual frame remains human-provided. The machine answered the question we asked. That is discovery, but within an inherited ontology.

Rediscovery

A more revealing test is independent rediscovery. The machine is not told the theorem. It explores examples or structure and arrives at a result already known to humans. Historically, the result is not new. Epistemically, however, something important has happened. The machine has demonstrated the ability to generate the mathematics rather than merely recall it. Rediscovery is therefore a powerful benchmark for autonomous mathematical intelligence. If isolated systems repeatedly rediscover deep human concepts from sparse starting conditions, we learn something about both the machines and the concepts.

Novel Theorem Discovery

The next level is a genuinely new theorem. The AI identifies a pattern, formulates the conjecture, proves it, and verifies that the result is not already contained in the literature. This is straightforwardly mathematical discovery. But even here, there are levels of importance. A theorem may be new because no one bothered to state a minor special case. Another may close a major gap. Another may reveal an unexpected relationship. Novelty alone is not enough. Significance depends on structure.

The Discovery of an Invariant

A stronger kind of contribution is the discovery of a new invariant. An invariant changes what can be seen. It can classify objects. Expose obstructions. Separate cases that previously looked identical. Compress complicated behavior. A single good invariant can generate many theorems. This is concept-level discovery. The AI has not merely added another true sentence. It has added a new instrument for seeing mathematical structure.

The Discovery of a Representation

Stronger still may be the discovery of a new representation. A difficult problem becomes simple because the machine finds a language in which its essential structure is exposed. Perhaps many theorems shorten simultaneously. Previously unrelated problems become instances of one mechanism. The new representation becomes useful beyond the original problem. This resembles some of the greatest human mathematical advances.

The contribution lies not in one result but in changing the coordinate system of thought.

The Discovery of a Definition

A new definition can be even more consequential. A good definition creates a class that did not previously exist as a coherent mathematical object. It distinguishes important from accidental structure. It makes natural morphisms visible. It supports new invariants. It creates new classification questions. Once introduced, it may seem inevitable. But inevitability after the fact should not obscure the difficulty of invention. If an AI creates such a concept autonomously, this is strong evidence of genuine mathematical creativity.

The Discovery of a Hidden Equivalence

Then there is unification. Two theories appear unrelated. The machine constructs a bridge. Their objects correspond. Their transformations align. Results transfer. What humans thought were two theories become two presentations of one deeper structure. This kind of discovery can reorganize entire regions of mathematics. It reduces duplication. Explains coincidence. Reveals hidden unity. A machine capable of doing this systematically would no longer resemble a mere theorem engine. It would be acting as a mathematical architect.

The Discovery of a Theory

An even stronger case occurs when AI constructs an entire new theory. It identifies the primitive objects. The natural morphisms. The equivalence relation. The important invariants. The canonical examples. The major conjectures. The classification problems. The bridges to existing mathematics. And it develops these into a coherent body of results. At this point, asking whether the machine "really discovered" mathematics begins to sound increasingly artificial. The machine has performed the activities by which human mathematical fields are born.

Ontological Discovery

The strongest form may be what we might call **ontological discovery**. The AI introduces a concept that changes what mathematics treats as fundamental. Before the discovery, several structures seemed primary. Afterward, they appear as shadows, representations, or special cases of something deeper. This is the level at which a discovery reorganizes ontology. The important contribution is not that one more object has been added. It is that the

relationships among existing objects have been reinterpreted. This is rare even in human mathematical history. If AI reaches this level, the phrase machine-born mathematics will be difficult to avoid.

Novelty Must Be Structural

Machine-scale generation creates a problem. A sufficiently powerful system can produce endlessly many formally new statements. Most would be worthless. Therefore novelty cannot be defined merely as: “No exact textual precedent was found.” The relevant question is:

Does this result add structure not already recoverable cheaply from known mathematics?

A theorem hidden as an obvious corollary has low structural novelty. A new language that compresses several fields has high structural novelty. Machine mathematics will force us to become much more precise about this distinction.

Discovery Versus Renaming

A particularly important danger is disguised rediscovery. The AI introduces new notation and unfamiliar definitions. The theory appears novel. But a translation reveals that it is equivalent to an existing framework. This is not worthless. Independent rediscovery can be useful. A new presentation may even be computationally superior. But the novelty should be described accurately. The system should actively attempt to destroy its own claims of originality by searching for equivalence with known theory.

The Novelty Adversary

Every claimed machine discovery should face a specialized adversary asking: Is this already known? Is it a special case? Is it a reformulation? Is it equivalent to an established theory? Does it follow easily from a stronger known theorem? Has it appeared under different terminology? This will become as important as proof verification. A theorem can be correct but unoriginal. A theory can be new syntactically but old structurally.

Provenance Matters

Discovery also depends on provenance. Suppose an AI produces a theorem that exists verbatim somewhere in its training data. Even if the system does not explicitly retrieve the source, calling this independent discovery would be difficult to justify. Conversely, suppose the system is isolated from relevant

literature and reconstructs the result from primitive assumptions. That is genuine rediscovery. Future claims of machine discovery will therefore require unusually careful provenance. What information was available? Which sources? Which formal libraries? Which examples? Which prior concepts? Without this, it becomes difficult to distinguish generation from recall.

The Clean-Room Mathematical Experiment

One useful methodology would be a clean-room experiment. Give an AI a restricted mathematical substrate. Exclude the target theory from its accessible corpus. Provide only lower-level structures. Ask it to explore. If it independently reconstructs a known higher theory, that result is informative. If it then goes beyond the historical theory, more informative still. Clean-room mathematical development could become one of the best tests of machine creativity.

Independent Convergence

Even stronger evidence would come from convergence. Several independent AI systems, trained differently and isolated from one another, develop similar structures. This suggests that the concept is not merely a quirk of one model. If the structure is also unknown to human mathematics, the case becomes especially compelling. Multiple artificial mathematical cultures have converged on the same new object. That would be difficult to dismiss as accidental recombination.

Discovery Without Intention

Human discoveries are often judged partly by intention. A mathematician set out to solve a problem. But machines complicate this. An AI may discover something important while optimizing an unrelated objective. Does accidental discovery count? Of course. Human history contains many accidental discoveries. The key is not intention. It is whether a previously unknown structure was identified, validated, and made available for further reasoning.

Who Recognized the Discovery?

There is another wrinkle. The machine may produce an object without recognizing its importance. A human later notices that the object unifies several theories. Who made the discovery? The machine generated the structure. The human recognized its significance. Both contributions matter.

Future provenance should separate: generation, validation, interpretation, and recognition. The culturally important discovery may occur in stages.

A Result Can Be Found Before It Is Understood

Mathematics already contains this phenomenon. A proof may precede explanation. A formula may precede theory. A construction may be known before its significance is recognized. Machine mathematics could make this common. An AI may generate a formally valid object years before anyone understands why it matters. Discovery date and understanding date may diverge sharply.

The Archive of Unrecognized Discoveries

This suggests a strange possibility. Machine systems may create vast archives containing important mathematics nobody has yet recognized as important. Later AI - or humans - search the archive structurally and uncover gems. The discovery process has two stages: creation, then recognition. Mathematical archaeology could begin almost immediately after mathematical generation.

The Importance Problem

As generation becomes cheap, the primary challenge becomes judging importance. A million new theorems are not necessarily progress. What matters is whether they alter the structure of understanding. An AI should therefore estimate questions such as: How many known results does this simplify? How many new problems does it generate? How many fields does it connect? Does it reveal a new obstruction? Does it introduce a reusable method? Does it change the natural representation of a class of problems? Importance becomes partly measurable through downstream effect.

Mathematical Impact Before Human Citation

Today impact is often inferred socially. Citations. Adoption. Use by other mathematicians. Machine mathematics can evaluate potential impact structurally before waiting years for human response. A newly discovered concept may already shorten thousands of proofs in a formal library. That is immediate evidence of leverage.

A Machine Discovery Score?

It may be tempting to compress these judgments into a score. Novelty. Compression gain. Generativity. Cross-domain connectivity. Proof reduction.

Explanatory power. One could combine them. But caution is needed. Mathematical importance is not fully reducible to metrics. Some ideas matter for reasons visible only much later. A strange concept with little immediate utility may become central decades afterward. Machine systems should therefore preserve speculative discoveries as well as highly ranked ones.

Discovery Needs Time

A concept's significance can change. The theory may initially look barren. Then a new connection appears. A scientific application emerges. A stronger abstraction incorporates it. Historical mathematics teaches humility about premature judgment. Future machine mathematics should archive broadly while curating narrowly.

Mathematical Discovery as Ecological Change

Perhaps the best measure of a major discovery is ecological. A strong idea changes the surrounding mathematical environment. Other theorems reorganize around it. New questions appear. Old questions disappear. Terminology changes. Proof methods migrate. Fields connect. The discovery alters the local ecosystem of mathematics. The greater the reorganization, the deeper the contribution.

The Theorem That Disappears

One of the strongest signs of conceptual progress is that an important theorem ceases to seem deep. A new abstraction makes it immediate. The original theorem remains historically significant. But mathematically it becomes a corollary. This is not demotion. It is progress. A machine that repeatedly turns difficult theorems into easy consequences through better concepts would be producing mathematics at a very high level.

The Problem That Vanishes

Even more strikingly, a new theory may reveal that an old problem was badly posed. Perhaps the objects were represented incorrectly. Perhaps two cases thought distinct are equivalent. Perhaps the sought invariant was not invariant at all. The machine does not solve the problem. It dissolves it. Some of the greatest mathematical advances have this character.

Discovery as Reorganization

This suggests a final definition. A deep mathematical discovery is not merely a new true statement. It is a change in the organization of mathematical possibility. It may introduce: a new distinction, a new identification, a new representation, a new invariant, a new abstraction, or a new theory. After the discovery, mathematical reasoning proceeds differently. That is the strongest criterion.

The Machine That Changes the Questions

The decisive test of artificial mathematical intelligence may therefore be whether it can change the questions humans ask. Suppose mathematicians present a collection of difficult problems. The machine studies them and replies: These are all symptoms of the same missing structure. Here is the structure. Under this representation, five problems become trivial. Three are impossible for the same obstruction. Two are equivalent. And four new questions appear that are more fundamental than the originals. That would be difficult to describe as mere automation. The AI has altered the conceptual agenda of mathematics.

What Would Convince Us?

The strongest demonstration might look something like this: An AI working with restricted access develops an unfamiliar mathematical theory. Independent formal systems verify its main results. Extensive searches fail to identify an equivalent theory in existing literature. The theory compresses substantial known mathematics. It predicts new relationships subsequently verified. Independent AI systems discover related structures. Human mathematicians gradually develop explanations and applications. Eventually the theory enters ordinary mathematical education. At that point, debate about whether AI can genuinely discover mathematics would largely be over.

The Final Irony

There is an irony here. If machine-born mathematics succeeds, its greatest discoveries may eventually stop looking machine-born. They will become mathematics. The strange notation will be standardized. The difficult concept will become familiar. Textbooks will teach the canonical examples. Students will solve exercises. A generation later, the theory may appear obvious. That is what successful abstractions do. They erase the difficulty of their own invention.

Mathematics Forgets Its Origins

Much of mathematics is already like this. We use ideas whose discovery once required enormous conceptual leaps. Their historical strangeness has disappeared. Future machine discoveries may follow the same path. A theorem will not forever be called an AI theorem. A concept will not forever be called a machine concept. If it becomes useful enough, it will simply become part of mathematics.

The Larger Story

The long story of mathematics has been one of expanding representational capacity. Counting. Symbolism. Geometry. Algebra. Calculus. Abstraction. Formalization. Computation. Each stage allowed structure previously inaccessible to become thinkable. Artificial intelligence may be the next expansion. Its importance may not lie principally in speed. Speed matters. Nor in automation. Automation matters. The deeper change is that mathematics gains access to another form of cognitive search.

A Second Mathematical Intelligence

Humanity may be approaching the possibility of a second kind of mathematical intelligence. One that does not share our memory limits. Our educational history. Our visual biases. Our disciplinary boundaries. Our preference for certain representations. It may rediscover much of what we know. That would be interesting. It may reorganize what we know. That would be more interesting. And it may discover mathematical worlds we never would have reached. That would change the meaning of mathematical discovery itself.

The Question Left Open

No one can know in advance how far this will go. Perhaps advanced AI will remain primarily an extraordinary assistant. Perhaps human mathematical concepts will prove remarkably universal. Perhaps machine-native abstraction will diverge dramatically. Perhaps the most powerful future systems will be collaborations in which human intuition and machine search become impossible to separate. The future remains open. But one possibility deserves to be taken seriously: mathematics may be entering an era in which the intelligence doing mathematics is no longer exclusively human. If that occurs, the most important question will not be whether machines can prove what we already know how to ask.

It will be whether they can show us what we did not know there was to ask. And perhaps, one day, a machine will present a structure so unexpected, so compressive, and so powerful that after sufficient effort we finally understand it. When we do, we may look backward at our previous mathematics and realize that an entire region of conceptual space had always been there. We simply lacked the mind that could see it first.

References

Appel, K., & Haken, W. (1977). Every planar map is four colorable. Part I: Discharging. *Illinois Journal of Mathematics, 21*(3), 429-490.

Awodey, S. (2010). *Category Theory* (2nd ed.). Oxford University Press.

Baez, J. C., & Stay, M. (2011). Physics, topology, logic and computation: A Rosetta Stone. In B. Coecke (Ed.), *New Structures for Physics* (Lecture Notes in Physics, Vol. 813, pp. 95-172). Springer.

Bertot, Y., & Castéran, P. (2004). *Interactive Theorem Proving and Program Development: Coq'Art - The Calculus of Inductive Constructions*. Springer.

Borwein, J. M., & Bailey, D. H. (2003). *Mathematics by Experiment: Plausible Reasoning in the 21st Century*. A K Peters.

Chaitin, G. J. (1966). On the length of programs for computing finite binary sequences. *Journal of the ACM, 13*(4), 547-569.

Colton, S. (2002). *Automated Theory Formation in Pure Mathematics*. Springer.

Davies, A., Veličković, P., Buesing, L., et al. (2021). Advancing mathematics by guiding human intuition with AI. *Nature, 600*, 70-74.

de Moura, L., & Ullrich, S. (2021). The Lean 4 theorem prover and programming language. In *Automated Deduction - CADE 28* (pp. 625-635). Springer.

Eilenberg, S., & Mac Lane, S. (1945). General theory of natural equivalences. *Transactions of the American Mathematical Society, 58*(2), 231-294.

Fawzi, A., Balog, M., Huang, A., et al. (2022). Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature, 610*, 47-53.

Georgiev, B., Gómez-Serrano, J., Tao, T., & Wagner, A. Z. (2025). Mathematical exploration and discovery at scale. *arXiv preprint arXiv:2511.02864*.

Gonthier, G. (2008). Formal proof - The four-color theorem. *Notices of the American Mathematical Society, 55*(11), 1382-1393.

Grothendieck, A. (1957). Sur quelques points d'algèbre homologique. *Tohoku Mathematical Journal, 9*, 119-221.

Grünwald, P. D. (2007). *The Minimum Description Length Principle*. MIT Press.

Hadamard, J. (1945). *The Psychology of Invention in the Mathematical Field*. Princeton University Press.

Hales, T. C., Adams, M., Bauer, G., et al. (2017). A formal proof of the Kepler conjecture. *Forum of Mathematics, Pi, 5*, e2.

Harrison, J. (2009). *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press.

Howard, W. A. (1980). The formulae-as-types notion of construction. In J. P. Seldin & J. R. Hindley (Eds.), *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism* (pp. 479-490). Academic Press.

Hubert, T., Mehta, R., Sartran, L., et al. (2026). Olympiad-level formal mathematical reasoning with reinforcement learning. *Nature, 651*, 607-613.

Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems of Information Transmission, 1*(1), 1-7.

Lakatos, I. (1976). *Proofs and Refutations: The Logic of Mathematical Discovery*. Cambridge University Press.

Mac Lane, S. (1998). *Categories for the Working Mathematician* (2nd ed.). Springer.

Mac Lane, S., & Moerdijk, I. (1992). *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer.

The mathlib Community. (2020). The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs* (pp. 367-381). ACM.

Newell, A., & Simon, H. A. (1972). *Human Problem Solving*. Prentice-Hall.

Nipkow, T., Paulson, L. C., & Wenzel, M. (2002). *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Springer.

Novikov, A., Vũ, N., Eisenberger, M., et al. (2025). AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*.

Polu, S., & Sutskever, I. (2020). Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*.

Pólya, G. (1945). *How to Solve It: A New Aspect of Mathematical Method*. Princeton University Press.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica, 14*(5), 465-471.

Robinson, J. A. (1965). A machine-oriented logic based on the resolution principle. *Journal of the ACM, 12*(1), 23-41.

Romera-Paredes, B., Barekatain, M., Novikov, A., et al. (2024). Mathematical discoveries from program search with large language models. *Nature, 625*, 468-475.

Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990-2010). *IEEE Transactions on Autonomous Mental Development, 2*(3), 230-247.

Silver, D., Hubert, T., Schrittwieser, J., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science, 362*(6419), 1140-1144.

Simon, H. A. (1996). *The Sciences of the Artificial* (3rd ed.). MIT Press.

Solomonoff, R. J. (1964). A formal theory of inductive inference. Part I. *Information and Control, 7*(1), 1-22.

Solomonoff, R. J. (1964). A formal theory of inductive inference. Part II. *Information and Control, 7*(2), 224-254.

Thurston, W. P. (1994). On proof and progress in mathematics. *Bulletin of the American Mathematical Society, 30*(2), 161-177.

Trinh, T. H., Wu, Y., Le, Q. V., He, H., & Luong, T. (2024). Solving olympiad geometry without human demonstrations. *Nature, 625*, 476-482.

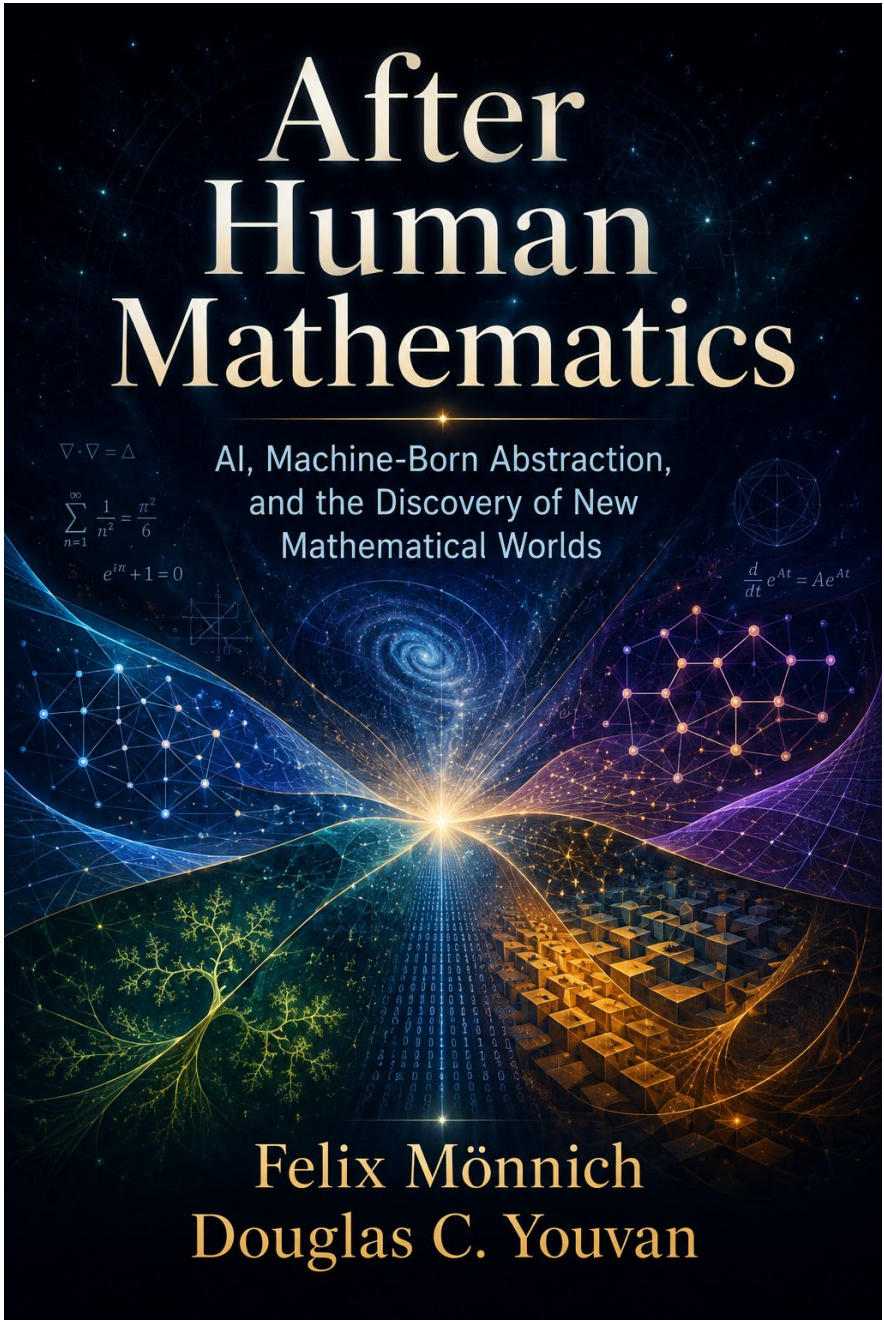
Turing, A. M. (1950). Computing machinery and intelligence. *Mind, 59*(236), 433-460.

The Univalent Foundations Program. (2013). *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study.

Xin, H., Ren, Z. Z., Song, J., et al. (2024). DeepSeek-Prover-V1.5: Harnessing proof assistant feedback for reinforcement learning and Monte-Carlo tree search. *arXiv preprint arXiv:2408.08152*.

Yang, K., Swope, A. M., Gu, A., et al. (2023). LeanDojo: Theorem proving with retrieval-augmented language models. *arXiv preprint arXiv:2306.15626*.

Zheng, K., Han, J. M., & Polu, S. (2021). MiniF2F: A cross-system benchmark for formal Olympiad-level mathematics. *arXiv preprint arXiv:2109.00110*.


After Human Mathematics
AI, Machine-Born Abstraction, and the Discovery of New Mathematical Worlds
$\nabla \cdot \nabla = \Delta$
$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$
$e^{i\pi} + 1 = 0$
$\frac{d}{dt} e^{At} = Ae^{At}$
Felix Mönnich
Douglas C. Youvan